A KNOWLEDGE-BASED APPROACH
TO ABNORMAL EEG SPIKE DETECTION

By

SEUNG-HUN PARK

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1990

ACKNOWLEDGEMENTS

The author would like to thank his supervisory committee for their guidance and assistance in doing this work. In particular, the author wishes to thank Dr. Jack R. Smith for his kind, responsible guidance, support, and encouragement. The author is deeply indebted to Dr. Jose C. Principe who helped make this work possible in all aspects.

The author would like to gratefully acknowledge the collaboration of Dr. Edward J. Hammond and Dr. Basim M. Uthman of the Department of Neurology at the V.A. Medical Center for spending their precious time in scoring the EEG. He is also grateful to Dr. Donald G. Childers, Dr. A. Antonio Arroyo, and Dr. Steven A. Reid for their kindness and helpful comments.

Appreciation is owed to Mr. David Jura and Ms. Mary Heines at Shands Hospital for their help in recording EEG data, and to Karla and Helen at Microtronics Inc. for lending the SAC machine. Appreciation is also owed to Russell Walters for his editorial help; to Chong-Tai Kim, Tae-Whan Yoon and Haan-Go Choi for their helpful discussion; to Gyo-Beom Chung, Chie-Teuk Ahn and Taeg-Won Jeong for their encouragement and moral support. The author also wishes to thank all his colleagues at the Computational Neuroengineering Laboratory.

Finally, special recognition is expressed to his wife Hye-Kyong and his two daughters Ye-Hyon and Ye-Jin for their love and patience, and to his parents and his in-laws for their commitment to education and guidance for many years that resulted in this dissertation.

This work was partially supported by grants from NINCDS Grant R01-NS19996 and NSF Grant ICI-8511857.

TABLE OF CONTENTS

APPENDICES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

A KNOWLEDGE-BASED APPROACH TO ABNORMAL EEG SPIKE DETECTION

By

SEUNG-HUN PARK

December 1990

Chairman: Dr. Jack R. Smith
Major Department: Electrical Engineering

The goal of this dissertation is to develop an automated multichannel EEG analysis system to assist the EEGer in prescreening epileptic patients. The design concentrates on the problem of detecting epileptic inter-ictal spikes. User-friendly, window-based visualization tools and a multichannel epileptic spike detection system are developed in this study to provide a visualization environment and a spike detection performance better than that of existing systems.

The user-friendly, window-based tools developed here enable the user to simultaneously visualize and manage the results of automated EEG analysis and multichannel EEG data. The new waveform detection algorithm presented here performs the structural analysis of characteristic line segments, obtained through a guided line segment search. This algorithm was implemented and gave results comparable with those obtained from one of the best automated methods in the detection of sleep EEG waveforms.

The spike waveform detector using this algorithm made few detections in non-epileptic EEG data. In epileptics, it detected most of epileptic spikes, but it generated false alarms due to epileptic sharp activities and normal EEG activities.

The knowledge-based contextual analysis model, which uses a hypothesis-confirmation process to simulate the EEGer's visual EEG analysis, was used to develop a knowledge-based system for screening out false positive detections generated by the spike waveform detector. The system eliminates most false detections due to normal EEG activities such as alpha, sigma, muscle and eye-movement artifacts. However, it still suffers from false positive detections mainly due to epileptic sharp activities. In two subjects with epilepsy, 72% and 63% of the epileptic spikes visually screened by two EEGers were detected by the system, with 2.68 and 2.89 false detections per minute, respectively. In two subjects with no epileptic spikes agreed by both EEGers, 0.02 and 0.33 false positive detections per minute were obtained.

CHAPTER 1
INTRODUCTION

Since its discovery in 1924, the electroencephalogram (EEG) has proved to be an important noninvasive diagnostic tool in neurological disorders such as epilepsy, in the diagnosis and localization of brain tumors, in comma assessment during intensive care hospitalization, and in the definition and study of anesthesia as well as of sleep stages [Kt83].

For a clinical examination, the EEG is amplified from various scalp locations via high-gain differential amplifiers which are usually AC-coupled to the scalp electrodes. The amplified signals are traced onto paper via a polygraph, and typically 8 to 16 channels of information are presented simultaneously to the clinician, each channel depicting the time history of the biopotential difference between two locations on the scalp (bipolar recording) or between one scalp locations and a reference elsewhere on the subject (referential recording). In some cases (e.g., brain surgery) the EEG electrodes are placed directly on the exposed cortex, and in various experimental procedures electrodes may be lowered to subcortical regions for more localized recordings.

The clinical applications of the EEG frequently involves a long-term monitoring, which necessarily produces a large amount of EEG data. For example, the paper tracing can be over 400 m long for an 8-hour sleep EEG recording [Sm87], and the EEG monitoring for prescreening the candidates who need surgery and for localizing their epileptic focus may have to be carried out over prolonged intervals of 24 or more hours to capture the patient's clinical seizures.

A traditional way of processing the EEG paper tracing is the electroencephalographer's (EEGer) visual interpretation. In the visual EEG interpretation,

the EEGer uses criteria and heuristics obtained from his years of training and clinical experiences as well as the definitions and guidelines officially recommended. It is a subjective process; there exist disagreements among EEGers. There also exist inconsistencies in the same EEGer [Gl86, Got86].

Since the computer was introduced, researchers have turned to the computer as a possible labor saving device which can substitute for a human EEGer by undertaking the human EEGer's routine visual interpretation. Another reason for automation is that the computer provides a consistent and objective means of data analysis. The computer is also used with the hope that its greater (than available with human visual analysis) resolving power will lead to the discovery of additional information not detectable by visual analysis [Got86, Sm87]. However, despite rapid advances in computer technology, achievements have been only in limited areas, such as sleep studies.

1.1 Statement of the Problem

One of the major contributions of electroencephalogram (EEG) has been its application in the clinical evaluation and diagnosis of epilepsy. Epileptiform EEG activity manifests itself in terms of sharp, "spiky" waves (EEG spikes) which are distinct from the background activity. It is the presence of these EEG spikes that alerts the clinician about the possibility of epilepsy in the subject under examination. In general, individual sharp EEG waves, not necessarily fulfilling the criteria for a spike, are of importance and may indicate an abnormal EEG [Kt83].

Numerous automatic methods have been developed for the recognition and quantification of interictal spikes, sharp waves, and spike-and-wave bursts. The aim of these methods is not to carry out a completely automated diagnosis. Instead, the methods have mainly concentrated on 3 aspects: (1) to automate the detection of events which have a largely unpredictable occurrence and are sometimes infrequent; (2) to quantify the occurrence of these events in order to improve the traditional visual and necessarily

subjective interpretation; (3) to refine the analysis of the EEG by extracting from it information which is not available from visual inspection [Got86]. In other words, it is intended that the computer is used as an aide that relieves the EEGer of some of the tedious tasks in the visual EEG interpretation, not as a final decision maker; the EEGer is still left with the final, important, diagnostic decision.

In order to be efficiently used in clinical practice, automatic methods should meet the following requirements. One is that the automatic methods must significantly reduce the amount of data the EEGer must deal with, without making excessive false detections. This means that the automatic methods should make as many true detections as possible, without increasing false positive detections. Another requirement is that the automatic methods should provide the information which can be directly used by the EEGer in making a clinical diagnosis. In other words, the clinical knowledge that has been accumulated from the visual EEG analysis must be applicable to the results of the automatic methods. Finally, the automatic methods should function either in real-time, or provide off-line, faster than real-time analysis of lengthy records so that they can ease the work-load of the EEGer. There is also a need to have an environment which enables the EEGer to easily visualize and manage the results of the automatic methods along with the EEG, and with which he can make a final, important, diagnostic decision.

1.2 Related Research

In this section, automatic methods developed previously for detecting epileptic spikes are briefly described. The automatic methods are broadly grouped into two categories: mimetic methods and parametric methods. In the mimetic methods, computer programs are made to do what we believe the EEGer does when reading an EEG: break down the EEG into waves and then identify the particular waves that fit the definitions of epileptic spikes [Got86]. The difficulty of implementing a mimetic technique is that the definition of epileptic spikes remains much too vague for encoding in a computer program.

On the other hand, the parametric methods do not require the precise definition of a spike. The following is the brief review of the two different types of methods, which is extracted from [Got86, Kt83].

1.2.1 Mimetic Methods

A detailed review of early automatic methods for detecting epileptic spikes from a single EEG channel by using either analog circuits or microprocessors was made by Gotman and Ktonas [Got86, Kt83]. The early methods mainly rely on the measurement of one variable, such as amplitude or sharpness, whether absolute or relative to the background. However, several authors have found it unsatisfactory to rely on only one variable. The distributions of amplitude and duration of spikes and of background waves were examined, but it was found that simple thresholds could not differentiate between the two because of a considerable overlap between the distributions [Ma76].

Complex methods had to be developed, relying on multiple thresholds and the measurement of several variables; the variables were not necessarily independent from each other; for instance, a wave with steep rising and falling phases is likely to have a sharp apex [Got86]. Goldberg et al. characterized the amplitude and duration of background waves in a first pass over the data, and spikes were detected in a second pass by measuring the amplitude of the high-pass filtered EEG (above 7 Hz) and using different thresholds depending on the content of the immediate context [Go73]. Smith determined the slope of the EEG by computing its first derivative; if the slope reached a given threshold, then the system looked for a slope of opposite direction, also reaching that threshold, occurring within the next few milliseconds. The slope was computed after low-pass digital filtering to eliminate spurious high values due to small amplitude fast activity, and the sharpness of the apex over a time interval was also taken into consideration to eliminate false alarms due to high-frequency noise [Sm74]. The second derivative relative to the background are measured, and several thresholds for wave duration and for the absolute amplitude of the

spikes were used [Fr80].

In the method of Gotman and Gloor the EEG was broken into segments (a segment being the section of EEG between a minimum and the following maximum, or vice versa) and the segments grouped in half-waves in an attempt to ignore small amplitude riding waves [Go76]. The amplitude of each half-wave was then measured relative to the average amplitude of the half-waves in the preceding 5 sec. The duration of each half-wave was then measured. If 2 adjacent half-waves fell within a set of thresholds for relative amplitude and duration, then the sharpness at the apex, relative to the background, was also measured. The thresholds were such that a wave of small relative amplitude had to be very sharp to be detected as a spike, whereas a wave of large relative amplitude did not require as sharp an apex for detection.

Gevins et al. utilized a powerful and elaborate digital computer system to detect spikes in 16 EEG channels at real-time speeds [Ge75, Ge76]. The system computed an average of the curvature of the EEG waveforms in the first part of the record to be examined, and a threshold for the instantaneous curvature was established for the isolation of waveforms which were sharp with respect to the background activity. Candidate waveforms for spike classification were subjected to a sequential series of tests based on heuristics of visual analysis. These heuristics included coincidence of sharp transients on adjacent electrodes, excessively low or high rate of occurrence, surface-negative polarity for reference montages and phase reversals for bipolar montages, association with a wave of low curvature, and likeness to previously selected sharp transients.

The last few methods described are quite complex and allow detection of a large proportion of the spikes present in an EEG. However, they still suffer from an excess of false detections due to non-epileptiform waves that fulfill all the detection criteria (eye blinks, sharp alpha activity, sleep spindles, K complexes, EMG activity and technical or movement artifacts).

Glover et al. developed a high-speed multichannel signal processing system which

detects epileptogenic sharp transients in the EEG [Gl86] and a knowledge-based system for the elimination of false positive detections [Gl89]. The system was claimed to make comprehensive use of spatial and temporal context information available on sixteen channels of EEG, EKG, EMG, and EOG as well as information learned as the detection process takes place. However, the system performance in clinical practice was not reported in detail.

As far as we know, none of the mimetic methods developed so far is currently being utilized in clinical practice, mainly due to an excess of false positive detections they produce. There is increasing awareness among investigators involved in automated epileptic spike detection that the solution to reliable detection, if there is one, lies in greatly increasing the context information utilized by the detection system [Gl89].

1.2.2 Parametric Methods

Parametric methods are based on the representation of a section of EEG by a small number of parameters. Rather than describing each wave, the parameters represent the statistical properties of the EEG during the section for which they are determined. Spike detection using this parametric representation is based on the principle that a spike is an unexpected or statistically improvable event. This concept was proposed by Lopes da Silva et al., who used the autoregressive filter as a parametric model [Lo75, Lo77].

In the parametric model, it is assumed that the EEG can be represented by the linear filtering of white, normally distributed noise. For a section of EEG, the filter coefficients are calculated so that the filtered noise and the EEG would have the same statistical properties. The original signal is then passed through the inverse of that filter: a white, normally distributed noise should be obtained. If a point extremely unlikely to occur in a normal distribution appears, it must correspond to a point in the original EEG that did not fit the statistical characteristics of that EEG: such a point was termed a nonstationarity by Lopes da Silva. When the filter was applied to the EEG of epileptic patients, it was found

that many epileptiform spikes were non-stationarities: this method could therefore be used for spike detection. When Lopes da Silva et al. recorded EEGs from the scalp and from underlying subdural electrode in epileptic patients, they found that many of the non-stationarities detected in the scalp, although not concurrent with a spike, occurred when a spike was present in the subdural EEG [Lo77]. This method could therefore act like a magnifying glass on the EEG, detecting events clearly correlated to epileptiform spikes but not identified as such by normal visual inspection.

Despite its promise, the method could not be widely used because there are many non-stationarities that are not related to epileptiform spikes (abrupt changes in EEG background, vertex sharp waves, EMG artifacts) [Got86]. In the absence of subdural electrodes it is impossible to decide with certainty whether a non-stationarity is epileptiform or not. A more detailed review of the parametric methods is available in [Ja85].

1.3 Objectives of this Study

Most of the existing automated methods for detecting epileptic spikes reportedly perform well on relatively short EEG records with little artifact and well-defined epileptic spikes. However, none have been able to approach the EEGer's ability to distinguish epileptic spikes from artifact or sharp but normal EEG [Gl89]. In particular, any of the methods developed so far cannot reliably process long EEG records (e.g., a night's sleep record) without reporting an excess of false positive detections due to non-epileptiform waves that fulfill all the detection criteria [Got86]. Thus, clinicians are reluctant to use the automated methods in clinical practice. In order for the automated methods to be used in clinical practice, the number of false positive detections must be drastically reduced.

The main goal of this study is to obviate the difficulty that hampers the clinical application of the automatic methods. In this study, two innovations are made to develop an automated EEG analysis system that assist the EEGer in clinical practice. One is to

design a more reliable automated system for epileptic spike detection, and the other is to create a computer-based environment in which the EEGer can easily visualize and manage the results from the automated detection system, together with the EEG.

In automated detection systems, the reliability is determined by the frequency of false positive and false negative errors; the detection system with more false positive and false negative errors is less reliable. In this study, two approaches are explored to raise the reliability. A new morphological waveform detection algorithm is designed, in an attempt to decrease false negative errors without increasing false positive errors. The other is an attempt to eliminate false positive errors through a comprehensive use of spatial and temporal context information. A distributed expert system model is proposed to help the system to analyze various contextual clues in a way similar to the EEGer.

The chapters that follow describe the design, implementation, and testing of the automated detection system based on the new morphological waveform detection algorithm and distributed expert system model. A workstation-based environment for visualizing and managing the results of the automated detection system simultaneously with the EEG is also developed.

1.4 Overview

This section presents a brief overview of the remainder of this dissertation. There are three important parts in this dissertation. The first part describes a new workstation-based environment which provides the EEGer with a means of visualizing and managing both the results of the automated detection system and the EEG. The second part describes a new morphological waveform detection algorithm, while the third part describes a distributed knowledge-based system model and various knowledge representation schemes suitable for different types of contextual clues and the EEGer's heuristics involved in the visual EEG analysis. Each of these parts is described in a separate chapter.

In this chapter the problem of detecting epileptic spikes and an approach to solve

the problem have been discussed. This chapter has also reviewed the previous related work.

Chapter 2 describes the interactive design and analysis tools for visualizing and quantifying multiple channels of data. It describes how the tools help engineers in designing and evaluating filters and detection algorithms for EEG waveforms.

In Chapter 3, a new search-oriented morphological waveform detection algorithm and a unified specification model for various EEG waveforms are described. It also illustrates how various sleep waveform detectors can be implemented using the algorithm. Experimental results are shown to evaluate the performance of the waveform detectors.

Chapter 4 describes a distributed knowledge-based system model and knowledge representation schemes suitable for spatial and temporal contextual clues and the heuristics used in the visual EEG analysis. A knowledge-based contextual analysis system for eliminating false positive detections is also described. A comparison between the results of the system and the visual scorings made by two certified human EEGers is provided to evaluate the performance of the overall system, which consists of the waveform detection system and the knowledge-based contextual analysis system.

Chapter 5 summarizes the main ideas of this dissertation, and suggest directions in which this work can be extended. In Appendix A, models for various EEG waveforms and their characteristic line segments are presented, while an overall task specification and several specialized task specifications for the knowledge-based contextual analysis system are shown in Appendix B.

CHAPTER 2
TDAT AND EVES — TOOLS FOR TIME DOMAIN EEG ANALYSIS

In this chapter, interactive design and analysis tools for displaying and quantifying multiple channels of data are presented. The tools allow one to easily visualize multiple data channels and simultaneously observe the effects of filters on the data and to evaluate signal detection algorithms. The software systems are designed for a workstation environment; they will find application in a variety of applications where one needs to simultaneously visualize multiple data channels. TDAT and EVES are being used for the design and evaluation of filters and detection algorithms for electroencephalogram (EEG) waveforms, and they are serving as a prototype of a paperless system to be used by electroencephalographers.

This chapter describes the general software structure of the two systems and illustrates many of the system features with examples

2.1 Introduction

The human electroencephalogram (EEG) is usually recorded from electrodes attached to the scalp using high-gain amplifiers which are usually AC-coupled to the scalp electrodes. The amplified signals are traced onto paper via a polygraph, which contains typically 8 to 16 channels. Since its discovery, the EEG has been used for the diagnosis of epilepsy, for trauma assessment, for sleep research, and for the analysis of higher brain function. The EEG is highly dependent upon the availability of high quality instrumentation, and almost from the beginning, automated methods of signal quantification have been applied. One of the primary goals is to help the electroencephalographer (EEGer) in the time-consuming task of quantifying a signal that

appears to the eye as a low information content back ground intermixed with either bursts of rhythmic activity with different frequencies (the EEG rhythms) or short transients of clinical significance (such as spikes). In spite of years of research to produce universal automated detection methods, success has been achieved only in specific areas. Accomplishments include automatic sleep staging with a high degree of accuracy [Sm87, Ku88]; counting spikes and wave complexes [Gl89, Ko88, Got86, Pr85, Ol83], and monitoring in intensive care units [Lo87]. However, clinicians still rely on visual analysis for clinical applications.

The human eye-brain can be trained to recognize ostensively defined patterns in multichannel EEG recordings. However, ostensive definitions are not readily disseminated. A description of a mental image by words is normally poor and lengthy. What is needed in clinical practice is a way of exploring the great pattern recognition capabilities of the human visual system and enhancing the efficiency of the visual data communication.

Computers can bring quantification to EEG analysis in the form of precise measurements (microvolt and millisecond precision), but at this time, they cannot always use the measured data to identify clinically significant features.

All these aspects lead us to approach the use of computers in EEG research from a slightly different angle. We are also researching the design of computer-based environments that will help the engineer in the design of better automated detection methods, and the medical doctor in the visual clinical assessment of multichannel EEG recordings.

An obvious role of the computer is the creation of environments where the EEGer can visualize the data (mimicking the paper output of the EEG polygraph) and which are enhanced with the power of computer-based measurement and manipulation (i.e., storage and retrieval of very large data bases), good report generation, and automated detection methods that can direct the EEGer attention to suspicious events. The environment

described in this chapter, which consists of Time Domain Analysis Tool (TDAT) and Event Viewing and Editing System (EVES), is a preliminary step towards this goal.

Many engineering problems require the detection of relatively brief phasic events which are visualized or defined in the time domain. Any filtering done on the data to enhance the detection may distort the temporal characteristics of the signal of interest. These efforts are often visually described by visually observing the data before and after filtering. We are faced with this particular need in the design of algorithms for the detection of phasic events in the electroencephalogram (EEG). The TDAT and EVES were developed to provide an interactive means of visualizing data in the time domain, accurately measuring temporal features, and to provide a means of visualizing effects of various types of filters and detectors. These tools can find a wide application in data analysis, particularly biomedical signal processing. This chapter describes the nature of the problem for which the systems were designed and provides examples of their use.

2.2 Computer-Based Environments for EEG Visualization

One of the goals of this study is to design a paperless workstation environment which replaces the multichannel polygraph now used for EEG monitoring. TDAT and EVES are serving as a prototype for this design. They are also serving as a workstation environment for implementing new filtering and detecting algorithms, and for visualizing the results.

Most of the visual information extracted from the EEG is related to recurring short-term spatio-temporal characteristics of the signal. A computer environment for monitoring EEG signals should display high quality multichannel EEG recordings, enabling the user to scale the several amplitude and waveform scrolling speed, zoom in on a particular segment of the record to perform amplitude and period measurements, navigate at will in the EEG record, extract and compare segments, and apply digital processing algorithms to

clean the signal from artifacts or help focus the clinician attention. The operator has similar capabilities when monitoring a recording on paper (polygraph).

In order to be useful to the clinician, such a tool must also be made user-friendly. The concept of user-friendliness is vague, but it is not limited to being able to physically interact with the computer (mouse, windows, etc.). Most importantly, the software should raise the level of human-machine interaction to the level of the human decision making. With conventional programming, the user must control the machine at a very low level (the level of the program commands). Clinical communication involves specific questions which are directly related to the patient and the clinical environment. One natural way to harmonize the user goals with the computer specifics is to create an object-oriented environment centered around the clinical issues. TDAT and EVES already incorporate some features of an object-oriented environment such as their structure, the translation of clinical functions into software processes (such as the display, the measurements, the analysis), and the communication between processes.

The EEGer can progress with TDAT and EVES by using almost exclusively the mouse. All the information is presented to the user in the form of environments (implemented as windows) with mouse sensitive areas (commands). Other aspects taken into consideration during software development are expandability and versatility. The computer is simultaneously used as a powerful number cruncher for digital signal processing algorithms and a high quality data storage/display system.

TDAT and EVES are not only useful to EEGers, but they are also used in our laboratory as key tools to design and validate automated detectors. The computer simultaneously serves as the simulation environment for implementing digital signal processing algorithms and for visualizing the application of such algorithms. The visual feedback considerably decreases the algorithm development time; TDAT also serves as a rapid prototyping medium for new ideas.

2.3 Human Assessment of EEG Data

TDAT and EVES were designed with the visualization and analysis of the EEG in mind. Key points were identified as follows:

• EEG interpretation involves scanning large amounts of data. This requires a large data storage medium in the computer. The user must be able to navigate through data at will, i.e. go forward and backward through the record. An epoch should be displayable with multiple time and amplitude scales. The system must provide an epoch finder, i.e. a visual cue indicating the current position in the record. Searches should be fast (less than one second).

• In EEG interpretation, the EEGer visually analyzes the record (typically 4, 8, or 16 channels). The computer display must closely mimic the EEG tracings obtained with a polygraph. Sometimes there is a need to zoom-in on a particular data segment and measure the amplitude and period of particular waveforms. Moreover, the EEGer may want to select particular portions of the record for further analysis and comparison. The signal analysis can benefit from filtering to attenuate unwanted activity (e.g. base line drifts and high frequency artifacts). Also, from the engineer's point of view, accurate measurements are important for quantifying the effects of filtering, or establishing thresholds for automated detection algorithms. The capability to compare different filters, or detection algorithms (like peak detection) is also very convenient.

• The EEGer makes a mark on a paper tracing for later reference, when a clinically significant event is detected. The computer must display the EEG events detected by the EEGer or automatic detection systems, synchronized with the EEG, and the capability of searching for, retrieving, and modifying some EEG events is also required.

• Finally, the EEGer needs to reproduce clinically relevant portions of the data to append to the patient record. The printouts should describe what the clinician sees of clinical relevance. The engineer also needs paper records for illustration and documentation purposes.

2.4 Overall System Architecture

The software is written so that the user can access and control, from the console, the different aspects of the data visualization/analysis. The main software modules are associated with the data search and display, and with data processing. In this section, we describe the architectures of TDAT and EVES.

2.4.1 System Architecture of TDAT

Versatility in data analysis/visualization is achieved with software modularization (influenced by the application), with the following data structure: Data existing in mass storage is not directly displayed on the console. Figure 2-1 shows the overall functional block diagram of TDAT. The stored EEG data is loaded into a Global Data Buffer (GDB) by the Database Manager, and the data in a Global Screen Buffer (GSB) are displayed on the Display Window or on one of Subwindows by using the Display Manager, which changes amplitude and time scales, zoom in on a specific event, and etc. While the GSB can be a copy of the GDB (direct display of EEG data), the Function Executer can, at the user command, modify the data stored in the GDB before they are sent to the GSB for display. These modifications are performed by applying the digital signal processing algorithms kept in the Function Library, which can be expanded at will by the user, provided the set of standard procedures discussed below are followed. The fundamental function routines, which implement template signal processing algorithms (such as FIR, IIR filters, waveform detection, etc.), can have up to 20 function units. The user specifies the algorithm parameters in the function units (e.g. a bandpass FIR filter with a 10 Hz center frequency and 5 Hz bandwidth to analyze alpha activity). The first block receives data from the GDB and puts it in intermediate buffers that can be mapped to the GSB and which can serve as the input to other signal processing modules. The fundamental signal processing units can be applied in cascade. With this flexible arrangement, the user can test and visually evaluate single or combinations of signal processing units.

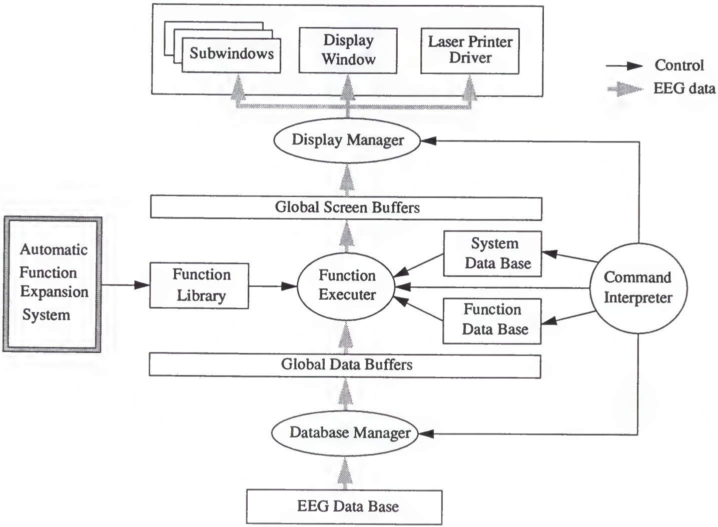


Figure 2-1. Overall functional block diagram of TDAT.

The currently active subject in EEG data base, scale information, the description of combinations of the fundamental function units to be applied to GSBs, the configuration of TDAT are kept in the System Database. The user interacts with the System Database through the main TDAT menu (see Figure 2-2). The user can access this menu from any point in the program modifying any parameter of the visualization/signal processing. Figure 2-2 exemplifies a session to set up the System Database, in which the user selected the data file "/home/data/r08pa02.dat" and the 30 second epoch by which the user can navigate the data file and eight display channels. The second channel of the Main Display (Screen Buffer 2) will display the output of a 60 Hz low-pass filter of which input is the first data buffer (Data Buffer 1). The user can choose other built-in functions by clicking the arrow in function set (a pop-up menu appears). The user can also register new function classes in this menu, or execute Unix commands in the window with the shell prompt. The Command Interpreter accepts the commands issued by the user, interprets and distributes them to appropriate modules.

The Function Library is expanded by writing code which conforms to its Input/ Output format and by executing the Automatic Function Expansion System (AFES). The AFES checks the code syntax and its compatibility with TDAT, compiles the code, and adds it to the Function Library. In addition, the AFES provides the capability to debug the logical validity of the code separately from the main TDAT software.

2.4.2 System Architecture of EVES

Event Viewing and Editing System (EVES) displays the EEG events detected by the EEGer or automated detection methods, synchronized with the EEG. Figure 2-3 show the overall functional block diagram of EVES.

EVES shares the EEG Data Base with the TDAT, but it has another data base for storing EEG events. An EEG event is specified by five attributes: the name and property of the event, the electrode derivations from which it takes place, and the time interval on

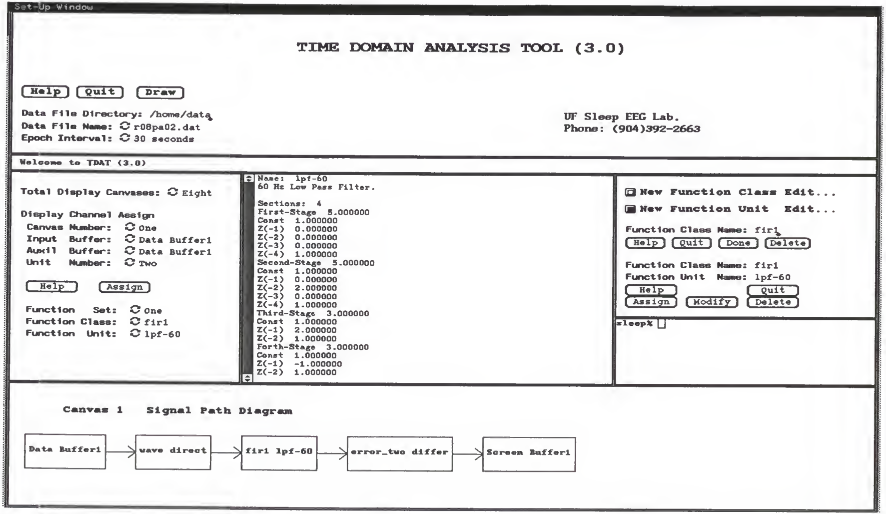


Figure 2-2. Set-up Window of TDAT.

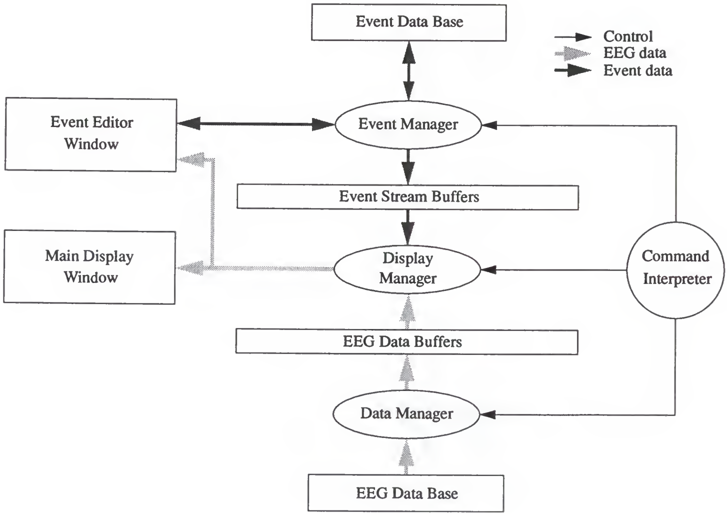


Figure 2-3. Overall functional block diagram of EVES.

which the event spans (the time interval of a time interval is specified by its start-time and end-time). In the Event Data Base, EEG events are stored according to the temporal order in which they occurred.

Upon the request of the command interpreter, the EEG data manager searches for the EEG segment to be displayed on the main display window. The command interpreter specifies the EEG segment by using the time interval of the EEG segment. The corresponding EEG segment is stored in the EEG data buffer until another EEG segment is loaded. The event manager searches for EEG events either having a certain type or occurring in the EEG segment contained in the EEG data buffer, and stores them in the event stream buffer. The display manager synchronizes the events in the event stream buffer and the EEG in the EEG data buffer, and displays them on the main display window.

The insertion, deletion and modification of EEG events in the event data base are achieved within the event editor window.

2.4.3 System Implementation

TDAT and EVES run under Unix on Sun Workstations, and most of the software is written in C. The implementation of the user interface is based on the Sun View System (Sun Visual/Integrated Environment for Workstations). Various items and subwindows provided by the Sun View System simplify building user-friendly, interactive, graphic, menu-driven interface. The full resolution (1152 * 900) of the Sun bit-mapped screen can be used for waveform display.

The EEG data are presently being collected in an IBM PC compatible computer with optical storage (8 bits, 250 Hz) and sent to the SUN 3 through an ethernet link (off-line).

2.5 Visualization

The identified tasks for EEG visualization are mapped into program environments. For easy interfacing, each environment corresponds to a window where the user selects, in command sensitive areas, the available commands. First, the visualization facilities of TDAT are described.

Data browsing is accomplished in the Main Display Window (shown in Figure 2-4). Depending on the sampling rate, the window can display up to 30 seconds of 8 data channels. The available commands are displayed in the screen's top row. The user simply selects the required function, and the program guides him/her through the steps by creating appropriate graphic menus. In the display window, the user can change the amplitude level and the time scale, move forward and backward in the data, or jump to a selected epoch by using the epoch finder.

The most frequently issued subwindow is the zoom window (shown in Figure 2-5), in which the user can closely scrutinize the EEG portions. The zoom window works as a software microscope, enlarging the time and amplitude scales without changing their ratio. This function is very important for precisely measuring wave amplitude, period and slope. The program automatically displays the measurements (uV and milliseconds) in the subwindow. The points used for the measurements are clearly labeled on the screen (See Figure 2-5). Also, since multiple channels can be plotted with respect to the same time reference, the synchrony of events across multiple channels is displayed with high resolution.

Power spectrum estimation is sometimes useful in estimating the dominant frequency component of relatively stationary, long-term events, or in evaluating noise-reduction algorithms. In the power spectrum window (shown in Figure 2-6), the user selects the segment of interest, and TDAT calculates and displays the estimated power spectrum. The power spectrum estimation is computed by averaging modified periodograms (Welch Method) [Op75]. When the selected data are not sufficient to implement the Welch

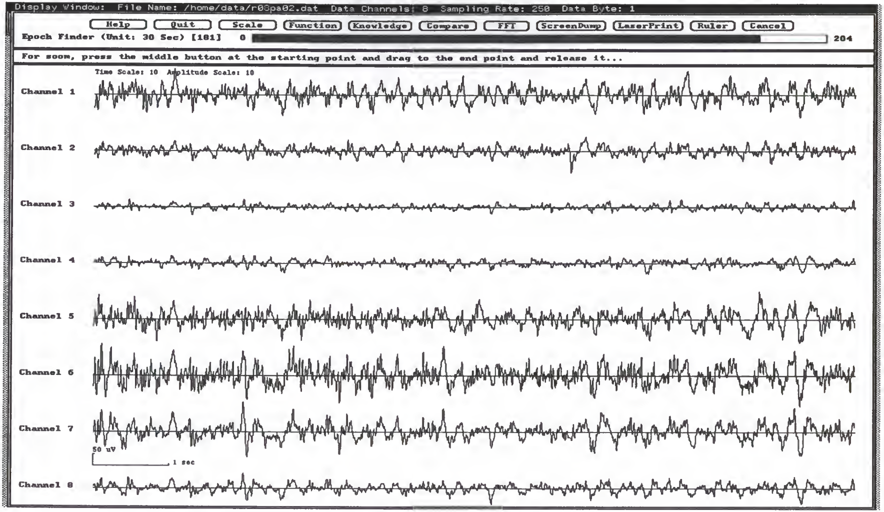


Figure 2-4. Main Draw window of TDAT.

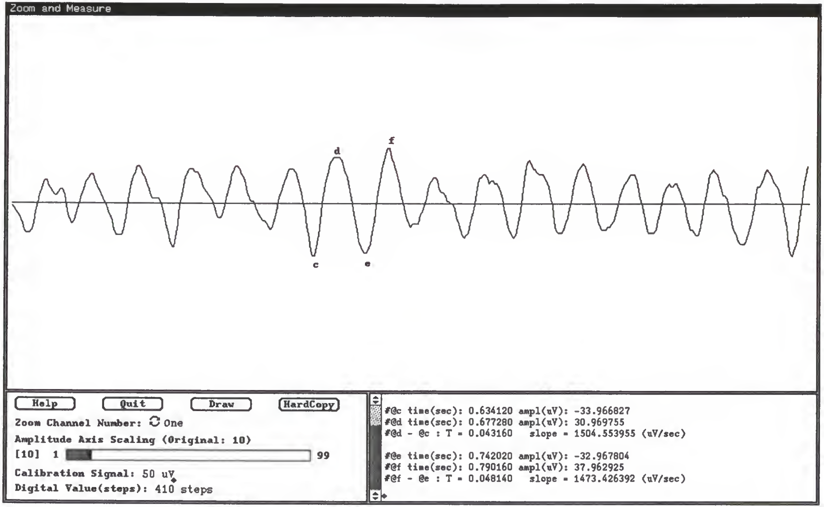


Figure 2-5. Zoom and Measure window of TDAT.

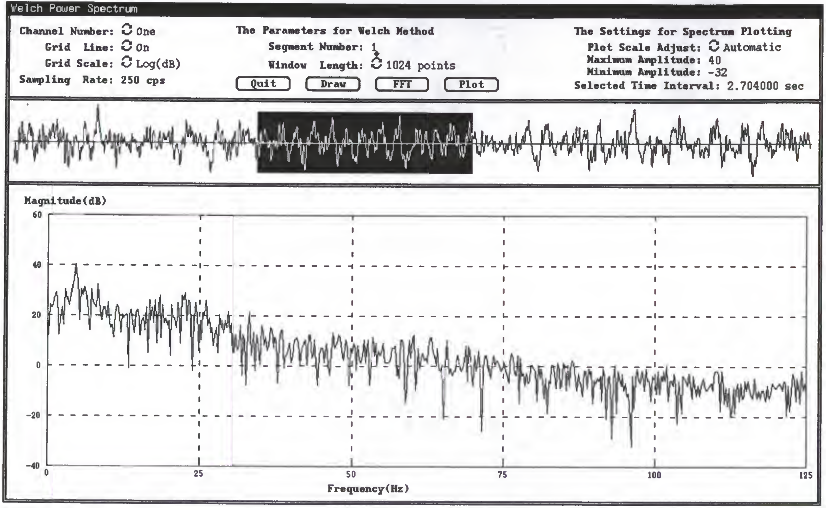


Figure 2-6. Power spectrum window of TDAT.

procedure, zero padding is performed. Long-term EEG monitoring produces large amounts of data. In some clinical applications such as the diagnosis of epilepsy, the clinician is interested in the portions of data having clinically meaningful events, which are scattered throughout the entire recording. The EEGer can select the clinically relevant segments using the data acquisition subwindow (shown in Figure 2-7). The clinically significant segments are extracted and stored sequentially in a data file. The program concatenates the segments, and the user can redisplay them in the main Display Window. This capability also simplifies the validation of detection algorithms by providing specific test patterns. The comparison window (shown in Figure 2-8) presents the differences between two display channels of the main display window. This is especially useful for evaluating a noise reduction algorithm, or for determining how much a filter routine suppresses unwanted detail. In this example, the differences between an original signal and the signal after low-pass filtering are shown. The comparison window can also accentuate asymmetries between EEG channels.

Another important visualization facility is a laser printer driver, which produces permanent archives of the computer screen. TDAT provides two kinds of hardcopy: screen-dump and laser-plot. Screen-dump creates a hard copy of the computer screen with a resolution of 100 dots per inch, while laser-plot produces a high resolution plot (300 dots per inch in most laser printers) of the EEG data. This is equivalent to the conventional polygraph paper tracing at a speed of 20 mm/sec. In spite of the existence of high resolution monitors, it is still difficult to achieve a computer display with the quality of a conventional polygraph paper tracing. The laser printer driver provides a quality that rivals ink jet EEG polygraph output.

Although EVES has many functions overlapping with TDAT, it also has unique features that can help the EEGer in clinical applications. Among them, the simultaneous display of the EEG and its clinically significant events helps the EEGer to locate portions of a clinical significance in reviewing a large amount of EEG data, over which the events

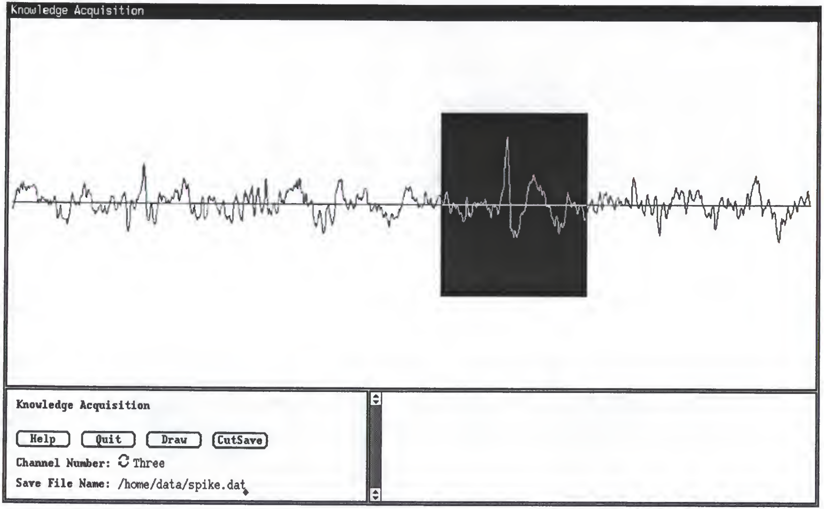


Figure 2-7. Data Acquisition window of TDAT.

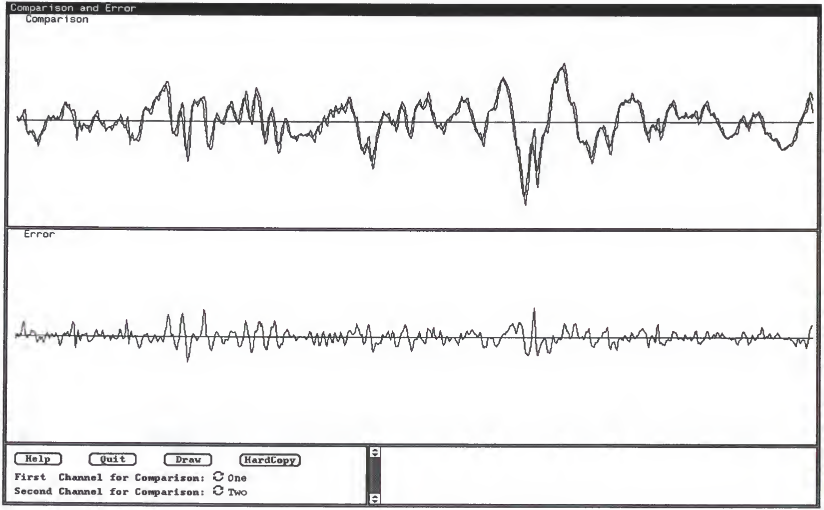


Figure 2-8. Comparison window of TDAT.

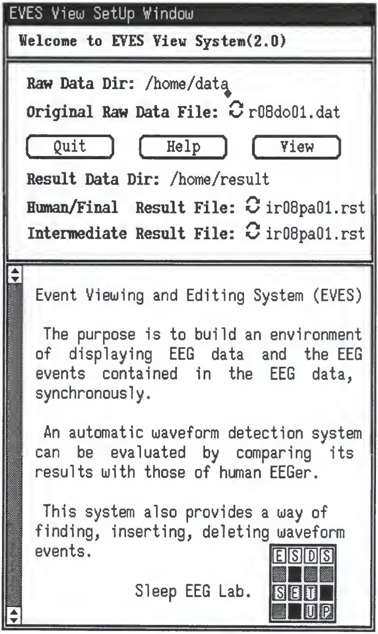


Figure 2-9. Set-up window of EVES.

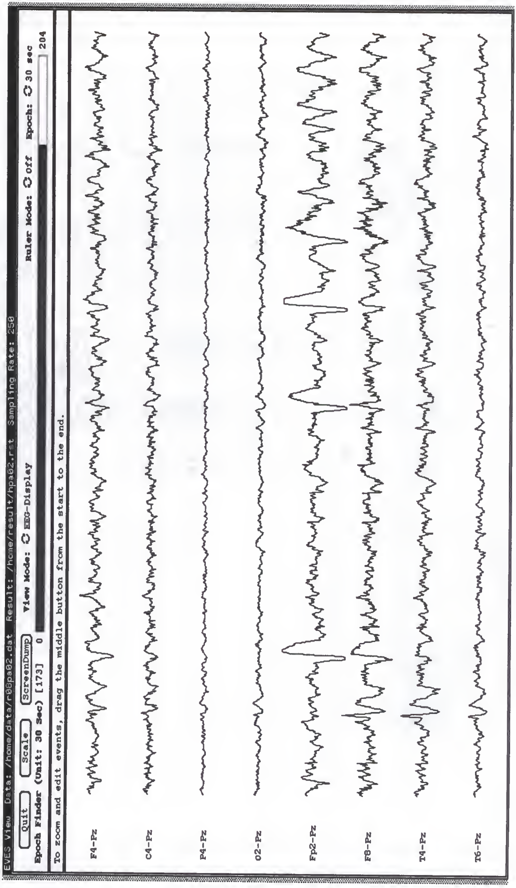


Figure 2-10. Main Display of EVES in EEG view mode.

are sparsely distributed. The Main Display Window has two display modes: EEG view mode and event view mode. In the EEG view mode, the EVES displays the EEG in a way similar to TDAT (see figure 2-10). However, in the event view mode, EEG events are marked with the initials of event names over the displayed EEG. In Figure 2-11, the EEG is displayed with two epileptic spike and wave complexes marked. EVES can be used in validating automated detection methods or in calculating man/machine agreements, since it can simultaneously display various events detected by two different EEGers; the upper black stripe in Figure 2-11 indicates the detections made by one EEGer, while the lower one indicates the detections by the other EEGer. Moreover, the Zoom and Measure window allows the user to insert, delete, find a specific event by dragging and clicking a mouse. Figure 2-12 shows a session of inserting a spike event into the archives file "/home/result/ hpa02.rst." Using this event editing facility, the EEGer can input his scoring to the event data base for later reference.

2.6 Discussion

Present day computer systems can be used for the time domain visualization of multichannel EEG data. In this sense, they can replace the currently used (paper) polygraph systems. However, computer-based systems are much more flexible. The operator can modify the data, quantify it, and enhance some of its features through digital signal processing. These characteristics are the basis for a much more productive interaction with the data. In computer-based systems the operator can examine and quantify in detail the waveform features. the computer provides a ready means for adding a broad class of signal processing algorithms (including linear phase filters), and for visualizing the effects of the filtering of the data. the computer provides a means of quantifying ostensively defined waveforms, random access to any segment, data archives and high-quality printouts.

Functionality in computer-based systems is achieved through programing. In this chapter we described a software architecture that was derived from the peculiarities of the

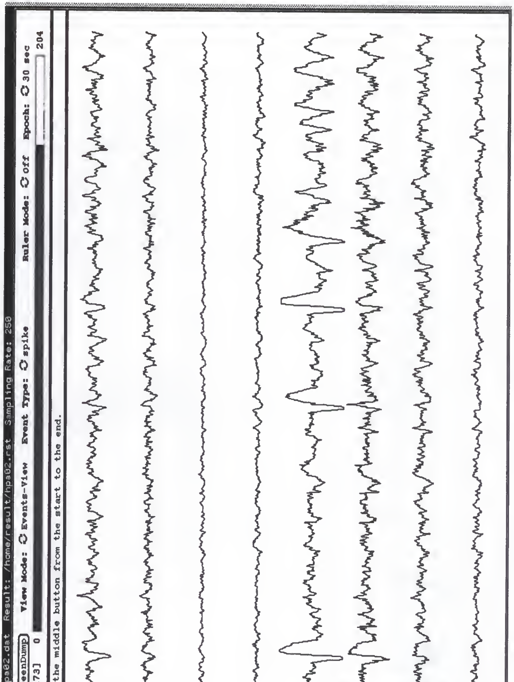


Figure 2-11. Main Display of EVES in Event view mode.

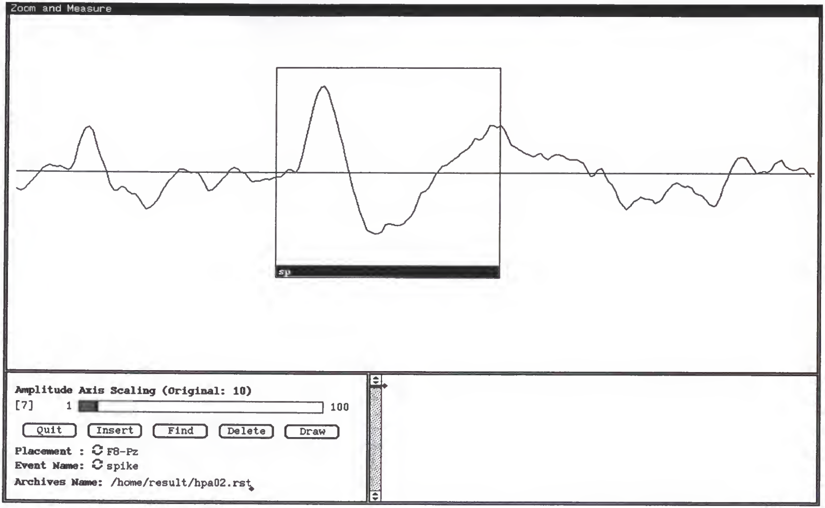


Figure 2-12. Event Editor window of EVES.

clinical assessment of EEG data and from the need to develop automated EEG waveform detectors. As the end result the EEGer has at his finger tips the added power of computer quantification, but his formal training can still be fully exploited because TDAT creates an environment centered around the signal and the operations to be performed on it. For the engineer interested in the design of waveform detectors, TDAT is also very helpful since it provides a ready means of prototyping and validating new detectors.

In engineering terms we would like to stress the open nature of the package and the graphic interface (both for menu selection and for building the signal processing algorithms). A current limitation compared to polygraph is the screen resolution. A computer screen with 1200 horizontal pixels can only provides an equivalent sampling rate of 40 samples per second if 30 seconds of data are to be displayed. The usage of TDAT is described in the user's manual of TDAT [Re89].

CHAPTER 3
SEARCH-ORIENTED WAVEFORM DETECTION

This chapter describes a new search-oriented waveform detection algorithm that can reliably detect EEG waveforms without using a priori signal conditioning. A unified format for specifying various waveform models is also provided. The search-oriented waveform detection algorithm consists of a search process for finding the characteristic line segments and a structural analysis process. The search process utilizes the graphical information of waveform morphology to limit the search space. Various numeric and symbolic structural parameters are defined and used in the structural analysis process.

Based on the waveform detection algorithm, a waveform detection system is developed to detect various EEG waveforms from multiple channels of EEG data. The system is configurable since its operation is determined, based on the waveform model specification provided by the user. In the experiments of detecting sleep EEG waveforms, the system performance is comparable with that of the reference system, which is one of the most reliable systems currently being used.

3.1 Search-Oriented Waveform Detection Algorithm

This section describes a new waveform detection algorithm for detecting EEG waveforms and a unified way of representing various EEG waveforms. The main idea of the new waveform detection algorithm is based on our observation that an EEG waveform is readily distinguished from other activities by its characteristic line segments together with their structural pattern. To detect an EEG waveform, the algorithm first searches for characteristic line-segments and stores them in a temporary finite queue with a limited queueing time. If the queue contains a sequence of line segments that follow any of the

given sequential patterns, a more elaborate analysis is performed to check the collective and structural features of the sequence. If the sequence meets collective and structural requirements provided by the waveform model, the EEG section that includes the sequence is reported to contain the EEG waveform.

The characteristic line segments are obtained through a search process guided by the graphical information of EEG waveform morphology, which is acquired from the statistical analysis of waveforms occurring in the EEGs of various subjects. Since the search process for detecting characteristic line segments does not require a priori signal conditioning, which could distort EEG waveforms in time domain.

3.1.1 Introduction

Automated EEG analysis involves two sequential tasks: the representation of the EEG and its clinical interpretation. The purpose of the representation is to convert the EEG data into a collection of clinically meaningful events, which can be manipulated by the computer. The EEG representation is used to make a clinical diagnosis in the subsequent clinical interpretation stage, together with other clinical clues obtained from different sources. In this chapter the discussion is confined to the issue "EEG representation."

Spectral (frequency) analysis has been the most frequently used method for describing the EEG. It is the process of transforming the data into the frequency domain. The main purpose of the transformation is to make some characteristics of the data more prominent in the frequency domain [Sm86]. The spectral (frequency) analysis has proved useful in some cases, but the resulting description cannot be directly used in the subsequent interpretation stage, since the medical and clinical knowledge which has been accumulated so far is mainly based on the human EEGer's visual scoring. Moreover, the spectral analysis is known to be inappropriate for describing aperiodic or irregular paroxysmal activities, which are commonly occurring in the EEG [Ku88].

Another popular approach, which we will call a mimetic approach, is to imitate what we believe human EEGer does during his visual analysis. In human visual EEG analysis the most important descriptors are waveforms occurring in the EEG. Thus, in the mimetic approach a special emphasis is given to the waveform detection, i.e., EEG waveforms are used as the EEG descriptors. Since the mimetic approach employs the same descriptors as those in the human visual EEG analysis, a large amount of medical and clinical knowledge that has been accumulated over time can be directly used in the subsequent interpretation stage.

In the mimetic approach, the EEG is first divided into multiple EEG segments which are likely to preserve important features. Several segmentation techniques have been developed for both waveform detection and data compression. The segmentation techniques used for waveform detection can be divided into two categories: point-oriented and line-oriented approaches. In the point-oriented segmentation approach, distinctive points are first chosen from the data, and EEG sections between the points are considered as primitive segments. Zero-crossing points [Ch88, Ku88, Sm78] and maximum and minimum points [Ko88, Fr80] are frequently used as the distinctive points. Although the point-oriented segmentation approach has some advantages in speed and simplicity of implementation, it often suffers from the existence of local points that appear to be distinctive, but in fact are of little global significance. Several preprocessing methods have been proposed to eliminate those local points, and some of them dramatically reduce the number of local points. But they introduce a different problem, unexpected waveform distortion produced by the preprocessing.

In the line-oriented segmentation approach the EEG is divided into multiple EEG segments which can be approximated by a straight line. Various pointwise error-norms are employed to control the approximation process. In this segmentation approach, the number of local points can be controlled by adjusting the threshold value of the pointwise error-norms. Some of the line-oriented segmentation algorithms use a piecewise linear

approximation to divide an initially taken, finite interval data segment. The algorithms using a piecewise linear approximation have yielded promising results in the areas of data compression and nonlinear filtering [Pa73]. However, they involve computation-intensive tasks such as the calculation of error norms and the piecewise linear approximation of individual data sections. Moreover, since the piecewise linear approximation is performed on initially taken, finite interval data, the algorithms are more suited to batch-processing.

In the mimetic approach, three different representation methods are used to represent collective and structural features of waveforms; they are numeric, symbolic and hybrid representation methods. Numeric representation methods employ a set of numeric parameters for representing collective and structural features of waveforms, while symbolic representation methods use concatenated symbols and syntactic rules to represent the collective property and structural features of waveforms, respectively. Both of representation methods are combined in hybrid representation methods.

The new waveform detection algorithm presented here employs a new line-oriented segmentation technique and uses numeric and symbolic parameters for representing the structural features of waveform morphology. The new line-oriented segmentation technique is implemented to be suited to real-time processing.

3.1.2 Search-Oriented Line Segment Detection

A variety of segmentation techniques have been attempted to represent EEG data for both data reduction and the application of syntactic pattern recognition techniques. In the waveform detection problems, most of these segmentation techniques were applied for the preliminary signal representation, without the explicit use of the knowledge of waveform morphology. However, the search-oriented line segment detection method presented here explicitly uses the morphological knowledge and is deeply involved in the main EEG waveform detection process as an important part, not as a separate preprocessing part.

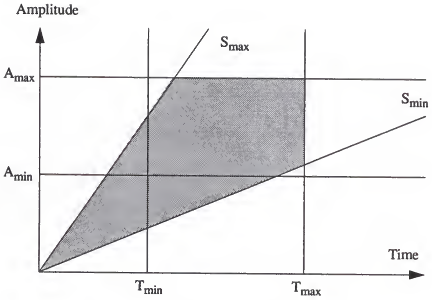


Figure 3-1. Search-continuation region

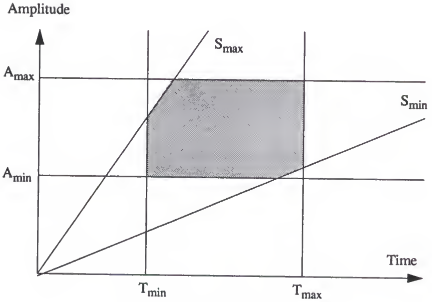


Figure 3-2. Valid line segment region

In the search-oriented line detection method presented here, a search process is guided by a set of numerical threshold values obtained statistically as shown in Figure 3-1. For simplicity of explanation, the time difference of two end-points of a line segment is denoted as the "time duration," and their amplitude difference as the "height." In Figure 3-1 and Figure 3-2, the origin of the graph corresponds to a common originating point of line segments, and the abscissa and the ordinate indicate the time duration and height of line segments, respectively. In the graphs, A_{min} and A_{max}, T_{min} and T_{max} denote minimum and maximum height threshold values, minimum and maximum time duration threshold values, respectively, while S_{min} and S_{max} denote minimum and maximum slope threshold values. These threshold values determine two regions: a search continuation region (the shaded area of Figure 3-1) and a valid line segment region (the shaded area of Figure 3-2). The search continuation region is used for limiting a search space, i.e., the search stops when a line segment is out of this region. The valid line segment region is used for finding a valid line segment, i.e., a line segment is valid if its terminating point lies within the region.

A line segment has two end-points, which are called an originating and a terminating data point, respectively. The data point used as a common originating point of all the line segments in the current search process is called a reference data point, while the terminating data point of the line segment currently being analyzed as a processing data point. The search process consists of four subprocesses: preliminary search for the first valid line segment, adjustment of search-continuation region, main search for candidate line segments, and selection of a representative line segment among candidate line segments.

Depending on the subprocess done before termination, the search process is classified into three different cases as illustrate in Figure 3-3: an initially failed search process (Figure 3-3(a)), a successful search process (Figure 3-3(b)), and a search process failed in the middle (Figure 3-3(c)). In Figure 3-3(a), E_i and E_{i+1} are taken as the reference and processing data points. Since the processing data point E_{i+1} fails to be within the

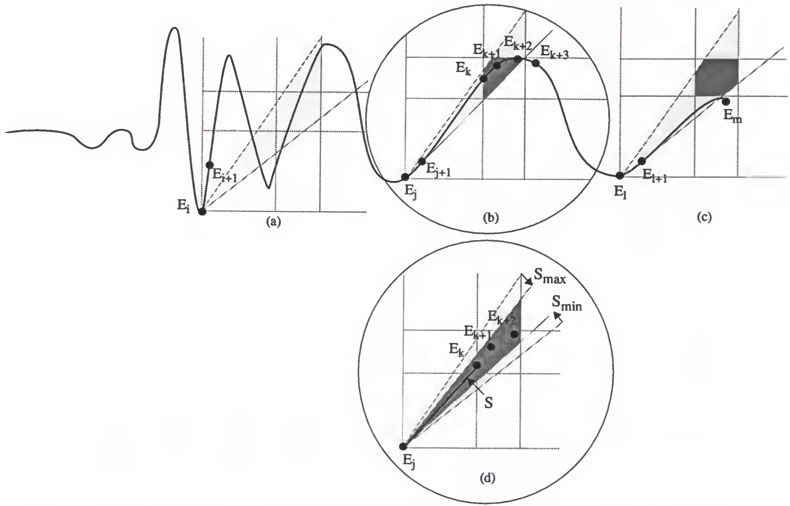


Figure 3-3. Examples of the search process. (a) Failed search process. (b) Successful search process. (c) Continued, but failed search process.(d) Adjustment of the search-continuation region after detecting the first valid terminating point.

search-continuation region, the search process is terminated. Therefore, a new search process would be initiated with the updated reference data point E_{i+1}. In Figure 3-3(c) E_l was taken as the reference point. Since the first processing data point E_{l+1} lies within the search-continuation region, the search process proceeds with updating the processing point until E_m is taken as the processing point. The search process terminates without reaching the valid line segment region, therefore a new search process is initiated. In the new search process, the data point E_{l+1} will be used as the reference point.

Figure 3-3(b) illustrates a successful search process, which results in the detection of a valid line segment. Initially, the search process starts with the reference data point E_j and the processing data point E_{j+1}, and the search process proceeds until the data point E_k is taken as the processing data point. In the current search process, the data point E_k is the first data point that successfully reaches the valid line segment region, therefore the search-continuation region is adjusted. Figure 3-3(d) illustrates the adjustment of the search-continuation region, in which minimum and maximum slope threshold values are updated to shrink the search-continuation region. New slope threshold values are calculated as follows:

$$S_{max,new} = \begin{cases} S + (S_{max,old} - S_{min,old}) \times \chi_{upper}, & S_{max,new} < S_{max,old} \\ S_{max,old}, & S_{max,new} \geq S_{max,old} \end{cases}$$

and

$$S_{min,new} = \begin{cases} S + (S_{max,old} - S_{min,old}) \times \chi_{lower}, & S_{min,new} > S_{min,old} \\ S_{min,old}, & S_{max,new} \leq S_{max,old} \end{cases}$$

where S is the slope value of the first valid line segment, and χ_{upper} and χ_{lower} are an upper and a lower contraction rate for shrinking the search-continuation region, respectively. Notice that the new search-continuation region is determined by the slope of the first valid line segment and the upper and lower contraction rates.

With the adjusted search-continuation region a new search process is initiated again for the same reference data point E_j. If the new search process successfully proceeds up to

the data point E_k and if E_k is still the first processing data point located within the valid line segment region, then the data point E_k is declared as a candidate terminating data point of a representative line segment in the current search process. On the other hand, if there were any data point out of the new search-continuation region before the search process gets to the data point E_k, the search process would be interrupted at that point, and the search process previously stopped would be resumed with the new processing data point E_{k+1} (the reference data point would be E_j). If that is the case, E_k would not be declared as a candidate terminating data point.

In Figure 3-3(d), the data points, E_k, E_{k+1} and E_{k+2}, are declared to be candidate terminating data points since the current search process reach the valid line segment region without violating the new search-continuation requirements. The search process continues until the processing data point fails to be within the valid line segment region. The data point E_{k+3} is not located neither within the search-continuation region, nor within the valid line segment region, and thus the current search process terminates.

Since the above search process found three candidate terminating data points, more analysis is performed to determine the representative line segment of the search process. From all the candidate terminating data points found above and the common reference data point, a candidate line segment set is generated. The representative line segment is selected from the candidate line segment set according to two criteria; a large amplitude criterion prefers a line segment with the largest absolute height, while a longest period criterion prefers a line segment with the longest time interval. Different criterion is employed depending on the type of the waveform under analysis. In our example, the largest amplitude criterion is used, i.e., the line segment between E_j and E_{k+2} is selected as the representative line segment of the search process.

When the search process is terminated with a representative line segment, the reference data point for a new search process is calculated differently. In Figure 3-3(b), the line segment that connects two data points E_j to E_{k+2} is selected as the representative line

segment of the search process just completed. Let t_j and t_{k+2} be the time of the two data points. The reference time $t_{r,new}$ of new search process to be performed right after is computed as follows:

$$t_{r,new} = \begin{cases} t_j + (t_{k+2} - t_j) \times \zeta, & (t_{k+2} - t_j) \times (1-\zeta) < T_{min} \\ t_j + (t_{k+2} - t_j) & (t_{k+2} - t_j) \times (1-\zeta) \geq T_{min} \end{cases}$$

where ζ is an overlap ratio of line segments. The reason for using the overlap ratio is to avoid the late initialization of the next search process, due to noisy segments. For example, if the longest time-interval criterion is used and if overlapping is not allowed, the next search process is always initiated at the data point between adjacent extreme data points, but not at the extreme data point.

3.1.3 Waveform Discrimination using Structural Features

In this study, each EEG waveform is distinguished by the characteristic line segments and their structural features. The structural features represent what kind of and how many line segments are involved and how they are structured to constitute the waveform. To use these structural features in automated waveform detection systems, the features must be represented in a manner that can be manipulated by computers. Here we introduce these structural features, describe them with both symbolic and numeric parameters, and present a way of representing various EEG waveforms in a unified fashion. Finally, we describe how to use these waveform models to detect EEG waveforms from a sequence of characteristic line segments.

3.1.3.1 Representation of Structural Features for an EEG Activity

There are two types of parameters involved in the representation of the structural features: symbolic and numeric parameters. Here, we exemplify what kinds of structural features are used and how they are represented. Figure 3-4 shows a line segment sequence, $\{L_1, L_2, L_3, ..., L_{n-1}, L_n\}$. The line segment L_i is identified by the originating and

terminating data points, which are represented by $(t_{i,1},a_{i,1})$ and $(t_{i,2},a_{i,2})$, and the type-name, which is denoted by "ls_i" (the symbols "t" and "a" indicate the time and amplitude of a data point, respectively).

The pattern of a line segment sequence is represented by concatenating the type-names of characteristic line segments according to the sequential order. To mark the separation of adjacent line segments, the symbol "•" is located between their type-names. In this example, the sequential pattern of the line segment sequence is denoted by "$ls_1 \cdot ls_2 \cdot ls_3 \cdot \ldots ls_{n-1} \cdot ls_n$." How adjacent line segments are connected is another important structural feature. The connectivity of adjacent line segments, L_i and L_{i+1}, is represented by two parameters: duty-ratio (δ) and time-duration (T) as follows:

$$T(L_i,L_{i+1}) = t_{i+1,2} - t_{i,1},$$

$$\delta(L_i,L_{i+1}) = \begin{cases} 1 - (t_{i+1,1} - t_{i,2})/T(L_i,L_{i+1}), & (t_{i+1,1} - t_{i,2}) \geq 0 \\ 1, & (t_{i+1,1} - t_{i,2}) < 0. \end{cases}$$

Some EEG waveforms such as alpha and sigma spindles are characterized by their prominent rhythmic activities. The symbolic sequential pattern parameter represents these rhythmic characteristics in a coarse manner. To represent these rhythmic characteristics more specifically and use them in the waveform detection, we introduce additional parameters. An duration-span-ratio and an amplitude-span-ratio contribute to describing how the amplitude and frequency are changing in a relatively short burst of rhythmic activity. A burst time-duration specifies how long the rhythmic activity lasts, and an average duty-ratio describes how sparsely all the characteristic line segments are joined. The four parameters are defined as follows:

$$\xi(L_i,L_{i+1}) = (t_{i+1,2} - t_{i+1,1})/(t_{i,2} - t_{i,1}),$$

$$\psi(L_i,L_{i+1}) = |(a_{i+1,2} - a_{i+1,1})/(a_{i,2} - a_{i,1})|,$$

$$T(L_i, \cdots L_j) = t_{j,2} - t_{i,1},$$

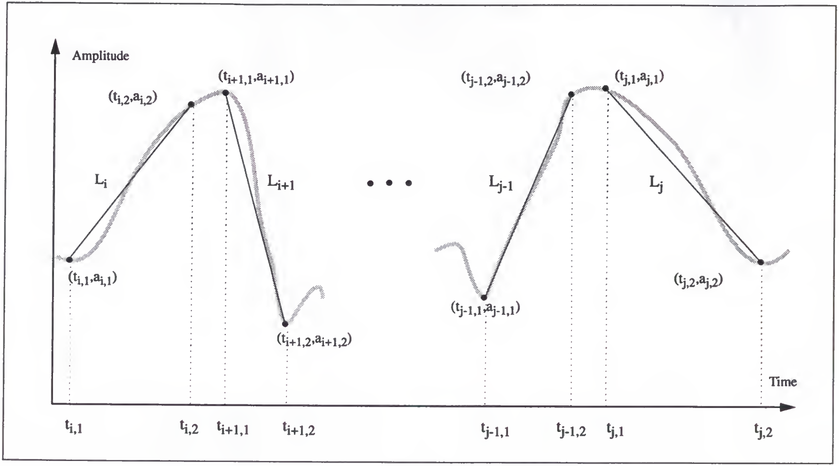


Figure 3-4. A sequence of consecutive line segments.

$$\delta(L_i, \cdots L_j) \quad = \quad \sum_{k=i}^{j-1} \delta(L_i, L_{i+1}) / (j - i - 1);$$

where ξ and ψ are the duration-span-ratio and amplitude-span-ratio of adjacent line segments, and $T(L_i, \cdots L_j)$ and $\delta(L_i, \cdots L_j)$ denote the burst time-duration and average duty-ratio of the line segment sequence $\{L_i, L_{i+1}, \cdots L_{j-1}, L_j\}$.

The structural feature parameters described build up a feature space in which line segment sequences with the exactly same structural features are mapped onto a single point, and line segment sequences having similar structural features are mapped onto clustered points. Since EEG waveforms are here represented by a properly structured line segment sequence of characteristic line segments, they can also be mapped onto points in the feature space.

If EEG waveforms were consistent from subject to subject, we would be able to construct a universal waveform model, which would mapped onto a single point in the feature space. In reality, EEG waveforms have a wide inter and intra-subject variation in waveform morphology. Thus if we employ a waveform model mapped onto a single point in the feature space, we may miss an abundance of authentic events whose shapes are slightly varying from the given waveform model. On the other hand, a loose waveform model mapped onto an excessively wide region in the feature space can introduce a lot of false detections.

To prevent such malfunctions, we must employ a waveform model that is mapped onto a proper region in the feature space. However, it is difficult to find the proper region, not only because of the ignorance of the mechanism that maps line segment sequences onto the feature space, but also because of the variations in waveform morphology. We mainly resort to human visual feedback to find a proper waveform model. In other words, we tune the feature parameter set by visually inspecting detection results, which we believe forms a proper region in the feature space. The region in the feature space can be specified by assigning upper and lower bounds of the feature parameters. Figure 3-5 (a) and (b)

illustrates the waveform and line segment models of an alpha activity, which consists of upper and lower bound threshold values of the feature parameters (these values were obtained empirically in our laboratory).

In the following two sections, we describe how these structural features are used to detect EEG waveforms in the waveform detection algorithm presented here.

3.1.3.2 Preliminary Sequential Pattern Filter

In traditional syntactic pattern recognition approach the sequential pattern of primitive elements is used as a main structural discriminator. However, in this work the sequential pattern of characteristic line segments is used as a preliminary discriminator to discard invalid sequences before a more elaborate analysis is carried out. Every waveform model contains a particular sequential pattern of characteristic line segments, which is represented symbolically as a concatenation of the corresponding line segment type-names. As shown in Figure 3-5 (a), alpha activity is specified by two different kinds of constituent line segments, "pcal" and "ncal", with the sequential pattern of "pcal·ncal·pcal·ncal·pcal·ncal·pcal·ncal." The models for the line segments "pcal" and "ncal" are defined in Figure 3-17(b). The search-continuation and the valid line segment regions are specified by Minimum-Period, Maximum-Period, Minimum-Amplitude, Maximum-Amplitude, Minimum-Slope, Maximum-Slope, and Slope-Direction. Selection-Policy specifies how to choose a representative line segment from multiple candidate line segments. Upper-Contraction-Rate and Lower-Contraction-Rate are used to adjust the search-continuation region.

A sequence of line segments detected in the search process is stored in a temporary queue with a queueing time limit. Every time a new line segment is entered, the time interval between the first and the last line segment is checked whether it exceeds a given time limit. If so, the first one is discarded from the queue.

// File Name — alpha.wave

Waveform-Type	:	*Multiple-Phasic*
Waveform-Name	:	*alpha*
Duration-Span-Ratio	:	*40%*
Amplitude-Span-Ratio	:	*20%*
Number-Of-Wave-Models	:	*2*

PatternDefinition

Waveform-Model-Name	:	*down-start-alpha*
Minimum-Total-Duration	:	*320 msec*
Maximum-Total-Duration	:	*572 msec*
Average-Duty-Ratio	:	*65%*
Sequential-Segment-Pattern	:	*ncal•pcal•ncal•pcal•ncal•pcal•ncal•pcal*
Minimum-Pair-Period	:	*<72><72><72><72><72><72><72> (unit:msec)*
Maximum-Pair-Period	:	*<156><156><156><156><156><156><156> (unit:msec)*
Minimum-Pair-Duty-Ratio	:	*<65><65><65><65><65><65><65> (unit:%)*

EndDefinition

PatternDefinition

Waveform-Model-Name	:	*up-start-alpha*
Minimum-Total-Duration	:	*320 msec*
Maximum-Total-Duration	:	*572 msec*
Average-Duty-Ratio	:	*65%*
Sequential-Segment-Pattern	:	*pcal•ncal•pcal•ncal•pcal•ncal•pcal•ncal*
Minimum-Pair-Period	:	*<72><72><72><72><72><72><72> (unit:msec)*
Maximum-Pair-Period	:	*<156><156><156><156><156><156><156> (unit:msec)*
Minimum-Pair-Duty-Ratio	:	*<65><65><65><65><65><65><65> (unit:%)*

EndDefinition

(a)

// Segment Name — pcal

Upper-Contraction-Rate	:	*8%*
Lower-contraction-Rate	:	*8%*
Minimum-Period	:	*20 msec*
Maximum-Period	:	*68 msec*
Minimum-Amplitude	:	*10 uV*
Maximum-Amplitude	:	*120 uV*
Minimum-Slope	:	*0.2 uV/msec*
Maximum-Slope	:	*5.5 uV/msec*
Slope-Direction	:	*Positive*
Selection-Policy	:	*Max-Amplitude*

Note) ncal is the same as pcal except that the Slope-Direction is negative.*

(b)

Figure 3-5. Alpha waveform specification. (a) Waveform model. (b). Line segment models.

Preliminary sequential pattern filtering begins with checking the type-name of the last line segment of the temporary queue. If the type of the last line segment is not matched with the last element of the given pattern template, the sequence will simply be rejected. On the other hand, the line segment sequence having the last line segment conforming to the given pattern template will be passed to the next step and will be tested whether it contains a set of line segments conforming to the sequential pattern template. The line segment sequence is said to be valid as long as it contains a set of line segments arranged according to the temporal order of the given sequential pattern template. From the temporary queue, the line segments matched with the sequential pattern template are extracted and form a new sequence of line segments, which will be called a candidate line segment sequence. For example, given a sequential pattern template "la·lb·lc·ld," the line segment sequence $\{L_1, L_2, L_3, L_4, L_5, L_6, L_7\}$ with the sequential pattern "la·lc·lb·la·lc·la·ld," is a valid line segment sequence, since it contains the last line segment with the type "ld," and it also contains other three different type line segments arranged in the order of the sequential pattern template "la·lb·lc·ld," i.e., the line segments L_1, L_3, L_5, and L_7 have the types "la," "lb," and "lc," and are arranged in the given order. Thus, the line segment sequence $\{L_1, L_3, L_5, L_7\}$ is a candidate line segment sequence. If it is failed to form a candidate line segment sequence, the structural analysis process will terminate, and new line segment search process will be initiated.

During the sequential pattern filtering, overlapping line segments of the same type are either merged or modified, depending on the type of the waveform being analyzed. This study classifies EEG waveforms into two distinct categories: single-phasic and poly-phasic waveforms. A single phasic waveform consists of two adjacent line segments of different types; delta waveform, which consists of two line segments of different types, is a typical example of single-phasic waveforms. On the other hand, a poly-phasic waveform consists of more than two line segments, such as the alpha waveform that consists of eight line segments.

Figure 3-6(a) illustrates the line segment merging process, which is performed in the detection of poly-phasic waveforms. All the overlapping line segments of the same type are merged into a single line segment as shown in the figure. Figure 3-6(b) illustrates the time and amplitude attribute modification performed in the detection of single phasic waveforms detection. The time and amplitude attribute of each line segment are adjusted, using a junction as a a reference. A junction refers to the point or interval at which two line segments of different types meet.

3.1.3.3 Numerically Represented Structural Feature Analysis.

To be a waveform, a candidate line segment sequence must be properly connected. Two structural parameters, time-duration T and duty-ratio δ, are involved to check whether the line segments are properly connected. The alpha waveform model in Figure 3-5 (a) contains three threshold values for the two structural parameters: minimum-pair-period $T_{min\text{-}pair}$, maximum-pair-period $T_{max\text{-}pair}$, and minimum-pair-duty-ratio $\delta_{min\text{-}pair}$. Figure 3-5(a) shows that a candidate line segment sequence must have the first two adjacent line segments that meet the requirement, $T_{min\text{-}pair} = 70$ msec, $T_{max\text{-}pair} = 150$ msec and $\delta_{min\text{-}pair} = 60\%$, to be a alpha waveform. Let the line segment sequence $\{L_0, L_1, L_2, L_3, L_4, L_5, L_6, L_7\}$ be a candidate line segment sequence of alpha waveform. The connectivity of the candidate sequence is tested by calculating the time-duration and duty-ratio of every adjacent line segment pair and checking whether it satisfies the corresponding connectivity requirements, which are specified by $T_{min\text{-}pair} \leq T(L_i,L_{i+1}) \leq T_{max\text{-}pair}$ and $\delta(L_i,L_{i+1}) \geq \delta_{min\text{-}pair}$ for $0 \leq i \leq 6$. For example, the connectivity requirements for the first adjacent line segments are specified by $70 \leq T(L_0,L_1) \leq 150$ and $\delta(L_0,L_1) \geq 60$. the candidate line segment sequence that has any pair that fails to meet the corresponding requirements is not properly connected.

A candidate line segment sequence properly connected undergoes a rhythmicity test. Four parameters were already introduced to represent the rhythmic features of line

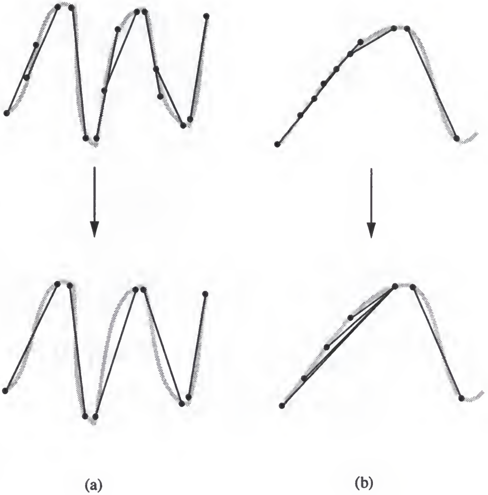


Figure 3-6. A line segment adjustment. (a) A line segment merging process in the poly-phasic waveform detection. (b) An attribute modification in the single-phasic waveform detection.

segment sequences. The waveform model presented here employs five threshold values for those rhythmic feature parameters to specify the rhythmic characteristics of EEG waveforms. For example, the rhythmic features of an alpha waveform are specified by five threshold values, $\xi_{threshold} = 0.5$, $\psi_{threshold} = 0.4$, $\delta_{min\text{-}average} = 80$, $P_{min\text{-}total} = 340$ msec and $P_{max\text{-}total} = 500$ msec, where $\delta_{min\text{-}average}$, $P_{min\text{-}total}$ and $P_{max\text{-}total}$ denote the minimum average duty-ratio, minimum-total-duration and maximum-total-duration of a line segment sequence (See Figure 3-5 (a)). The rhythmicity requirements are specified as follows: (let the line segment sequence $\{L_0, L_1, L_2, L_3, L_4, L_5, L_6, L_7\}$ be a candidate line segment sequence.)

$P_{min\text{-}total} \leq T(L_0, L_1, L_2, L_3, L_4, L_5, L_6, L_7) \leq P_{max\text{-}total}$,

$\delta(L_0, L_1, L_2, L_3, L_4, L_5, L_6, L_7) \leq \delta_{min\text{-}average}$,

$\xi(L_i,L_{i+1}) \geq \xi_{threshold}$ and $\psi(L_i,L_{i+1}) \geq \psi_{threshold}$, for $0 \leq i \leq 6$.

If the candidate line segment sequence meets the above requirements, the sequence will be labeled as alpha, and its occurrence time and duration are recorded together with the electrode placements where it occurs.

3.2 EEG Waveform Model Specification

Using the structural features described above, the morphology of EEG waveforms are specified in a unified way. Two different types of models are involved to specify the morphology of an EEG waveform. A waveform model specifies what kinds of characteristic line segments are involved and how they are structured as shown in Figure 3-5(a), while a line segment model specifies all the parameters needed in the search process for a certain type of characteristic line segments, as shown in Figure 3-5(b). In Figure 3-5(a), two different types of characteristic line segments, pcal and ncal, are involved in the alpha waveform specification.

In the waveform model, the attribute "Waveform-Type" identifies the waveform type, either single-phasic or poly-phasic, and the attributes "Duration-Span-Ratio" and

"Amplitude-Span-Ratio" specify the overall rhythmicity of the waveform activity. The attribute "Number-Of-Patterns" specifies how many different sequential patterns are included in the waveform model. In the pattern definition, the attribute "Pattern-Model-Name" identifies the pattern model, and other structural requirements previously described as specified.

In the line segment model, seven threshold values, "Minimum-Period," "Maximum-Period," "Minimum-Amplitude," "Maximum-Amplitude," "Minimum-Slope," "Maximum-Slope," and "Slope-Direction," determine the search-continuation region and the valid line segment region. The attributes "Upper-Contraction-Rate" and "Lower-Contraction-Rate" are used for the adjustment of the search-continuation region. The attribute "Selection-Policy" specifies the criterion for selecting a representative line segment from multiple candidates. As an example, the spike waveform model is shown in Figure 3-7. A spike waveform consists of two different characteristic line segments, "p1sp" and "n2sp." The models for the line segments are shown in Figure 3-8.

3.3 Development of Object-Oriented Waveform Detection System

Based on the search-oriented waveform detection algorithm and waveform model described above, an automated system for detecting various waveforms from multiple channels of EEG data was developed. The waveform detection system is configured and operates differently according to the waveform model provided by the user at run-time. It is configurable. The object-oriented concept is used for designing and implementing the waveform detection system.

3.3.1 Objects in the Waveform Detection System

The object-oriented development of a waveform detection system for detecting various waveforms from multiple channels EEG data involves six different classes of objects; the six classes are model class, event class, detector class, task control class, input

```
// File Name — spike.wave

Waveform-Type                    :    Single-Phasic
Waveform-Name                    :    spike
Duration-Span-Ratio              :    15%
Amplitude-Span-Ratio             :    30%
Number-Of-Wave-Models            :    1

PatternDefinition
 Waveform-Model-Name             :    upward-spike
 Minimum-Total-Duration          :    32 msec
 Maximum-Total-Duration          :    110 msec
 Average-Duty-Ratio              :    70%
 Sequential-Segment-Pattern      :    p1sp•n2sp
 Minimum-Pair-Period             :    <32> (unit:msec)
 Maximum-Pair-Period             :    <110> (unit:msec)
 Minimum-Pair-Duty-Ratio         :    <70> (unit:%)
EndDefinition
```

Figure 3-7. The spike waveform model specification.

```
// Segment Name — p1sp

Upper-Contraction-Rate           :    6%
Lower-Contraction-Rate           :    6%
Minimum-Period                   :    12 msec
Maximum-Period                   :    40 msec
Minimum-Amplitude                :    650 uV
Maximum-Amplitude                :    5000 uV
Minimum-Slope                    :    3.5 uV/msec
Maximum-Slope                    :    25 uV/msec
Slope-Direction                  :    Positive
Selection-Policy                 :    Max-Amplitude


// Segment Name — n2sp

Upper-Contraction-Rate           :    12%
Lower-contraction-Rate           :    10%
Minimum-Period                   :    20 msec
Maximum-Period                   :    70 msec
Minimum-Amplitude                :    200 uV
Maximum-Amplitude                :    600 uV
Minimum-Slope                    :    2 uV/msec
Maximum-Slope                    :    20 uV/msec
Slope-Direction                  :    Negative
Selection-Policy                 :    Max-Amplitude
```

Figure 3-8. Line segment models for p1sp and n2sp.

manager class and output manager class. Each class is characterized by its attributes and encapsulated procedures (called methods). We first identify the classes involved in the object-oriented development of the waveform detection system by describing their attributes and methods, and then describe how objects are instantiated from the classes and organized.

3.3.1.1 Model Class Objects

The search-oriented waveform detection algorithm consists of the line segment detection process and the structural feature analysis of the resultant line segments. In the line segment detection process, the line segment model, which specifies a search-continuation and a valid line segment region, provides a guideline information for determining when to stop and when to make a detection alarm; the waveform model holds the important information needed for performing the structural feature analysis. The two models are implemented as two different classes of model objects.

A line segment model class is characterized by its eleven attributes that specify the search-continuation region and the valid line segment region; they are minimum and maximum threshold values for specifying an amplitude, time, and slope range of both regions, and lower and upper contraction ratios for the adjustment of the search-continuation region. The line segment model class also encapsulates methods for reading its attributes from a corresponding configuration file while instantiating its objects and for modifying the attributes. The line segment model specification file is named after the line segment name with a "seg" extension. For example, the file "pcal.seg" stores the specification of the line segment "pcal." The word "instantiation" indicates the process of making a new instance of a class, i.e., the creation of a new object of a class.

A waveform model class has nine attributes: sequential-segment-pattern, minimum-pair-duty-ratio, minimum-pair-period, maximum-pair-period, minimum-average-duty-ratio, minimum-total-duration, maximum-total-duration, duration-span-

ratio, and amplitude-span-ratio. The waveform model class also encapsulates methods for reading its attributes from a corresponding specification file while instantiating its objects and for modifying the attributes. The waveform model specification file is named after the waveform name with a "wave" extension. The file "alpha.wave" stores the specification of the alpha waveform.

3.3.1.2 Event Class Objects

During the waveform detection process, two different types of events are created and manipulated. The first is a line segment event, which is generated during the search-oriented line segment detection process. The second type of event is a waveform event, which is generated when a candidate line segment sequence successfully meets the numerically represented structural requirements.

Both types of events have time information in common, which is represented by the start-time and end-time of the events. To implement these two events as two subclasses of a single superclass, we define a time event class as a super class. The time event class has two attributes: start-time and end-time. The time event class also encapsulates a variety of methods which can deal with the time information: "start-before," "start-same," "start-after," "end-before," "end-same," "end-after," "equal," "include," "overlap," "Intersection," and "Union." Additional methods for queue manipulation are also encapsulated.

A line segment event class is specialized from the class "time event class" by adding more attributes and methods. Since the line segment event class is a subclass of the time event class, it inherits all the attributes and methods of the time event class. The line segment event class has three additional attributes: "segment-type," "start-amplitude" and "end-amplitude". A waveform event class is also specialized from the class "time event class," thus it also inherits the attributes and methods of the time event class. In addition, the waveform event class has more attributes and methods for manipulating different

waveform occurrences and their time and placement information: the attributes "waveform-type," "waveform-name," "electrode-placements" and the methods ""equivalence," "is-same-type," and "at-same-placements."

3.3.1.3 Detector Class Objects

The search-oriented waveform detection algorithm is implemented by hierarchically structured detector class objects. They are line segment detector class objects at the low level and waveform detector class objects at the high level as shown in Figure 3-9 (the circles denote objects and the rectangles within a circle denote function units). The function units in an object are implemented as the methods of the object. The two detector class objects are composite objects since they contain other objects as parts; every line segment detector class object contains a selection queue and a line segment model class object, and the waveform detector class object contains a line segment queue and waveform model class objects. Thus, the detector class objects are specified by a class describing the part classes and their interconnections.

A set of line segment detector class objects located at the low level detects characteristic line segments from the same EEG channel; each line segment detector class object detects a particular type of characteristic line segments. The line segment detector class encapsulates several methods. Among them, the methods "search-segment-at" and "select-a-segment" performs two main tasks of the search-oriented line segment detection process, i.e., searches for a valid line segment from an EEG data stream and selects a representative line segment from multiple candidates in the selection queue.

A selection queue, which serves as one of the part objects of the line segment detector class object, stores candidate line segments detected in the currently ongoing search process. The selection queue is an object of a doubly-linked list of line segment event objects, which has two attributes that indicate the head and tail of the list and seven

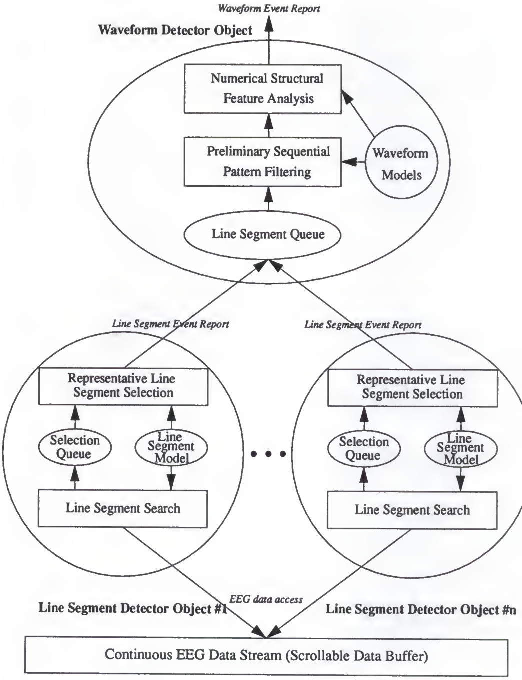


Figure 3-9. The organization of waveform and line segment detector objects for waveform detection.

encapsulated list processing methods: "append," "chronicle," "affix," "detach," "merge," "consolidate," and "tell-queue-length."

A high level waveform detector class object, in charge of the sequential pattern filtering and structural feature analysis, consists of two function units and three part objects: a line segment queue object, a candidate sequence queue, and a waveform model class object. A numeric structural feature analysis and a preliminary sequential pattern filtering are the two function units, each of which is implemented as a set of methods. The methods "waveform-matching" and "match-segment" perform the preliminary sequential pattern filtering, while the methods "detect-waveform-at" and "are-balanced" perform the structural feature analysis. The line segment queue stores representative line segments from the low level line segment detector objects during the preliminary sequential pattern filtering, while the candidate sequence queue stores a candidate line segment sequence, which passed in the preliminary sequential pattern filtering. The queues are all the same class as the selection queue.

3.3.1.4 Input /Output Manager Class Objects

The main goal of most EEG waveform detection systems is to extract clinically significant waveform events from an ongoing EEG data stream. This study used the EEG data already digitized and stored on a hard disk, which do not have to be processed in real-time. Nevertheless, all the modules of the system are built on the basis of real-time processing concepts in order to easily convert to a real-time processing system without major modification. However, in order for the real-time based modules to operate in a batch-style environment, the data should be supplied in a way to what would be obtained from A/D converters, i.e., the data must be given one sample at a time.

An input manager class contains a number of scrollable data buffers, each of which stores the data from a particular EEG channel. Methods for reading EEG data from a data file on a hard disk and for loading them into the scrollable data buffer are also encapsulated

in the input manager class. The input manager class has the attributes for holding the data file name, the sampling-rate, the number of bits per sample, the number of data channels, and the amplitude of a reference signal for calibrating various amplitude dependent system parameters. These attributes help to properly read a data file.

The scrollable data buffers act as a sliding window through which the least amount of the EEG needed for detecting EEG waveforms can be seen. As described above, the search-oriented line segment detection units and the input manager unit are implemented as independent objects of different classes. To reduce the amount of interactions between these independent objects, EEG data segments in the scrollable buffer are specified by the starting memory location and the data segment size. To access a proper EEG segment by the two parameters, the physical contiguity of data storage must be preserved, i.e., data samples must be stored in physically contiguous memory cells. For example, if a data sample is stored in the memory cell of which the address is "p", then the next data sample must be stored in the memory cell of the address "p+1."

The scrollable data buffer achieves the physical contiguity of data storage required by keeping two overlapping regions at both sides of the buffer; the regions are overlapping in the sense that one keeps a copy of the other. Figure 3-10 shows the scrollable buffer with four fixed and two moving pointers. A traditional circular buffer implemented by a stack of limited memory cells and a moving index pointer has an intrinsic drawback of losing the physical contiguity of data at the extreme end of the buffer. However, the scrollable data buffer maintains two moving pointers, a reference pointer and an input pointer, in order to keep the physical contiguity of data at the extreme end of the buffer.

In Figure 3-10, the input pointer (IP) indicates the memory location into which a new data sample will be loaded, and the reference pointer (RP) indicates the memory location on which data access is performed. After a new data sample is loaded, IP is updated as follows:

$$IP_{new} = \begin{cases} IP_{old} + 1, & IH \leq IP_{old} < PT \\ IH, & IH = PT. \end{cases}$$

If $RT \leq IP_{old} < PT$, auxiliary input pointer IP_{aux}, which is equal to IP_{old} - RT, is used to keep the same copy of the data sample in the front side overlapping region. The work space that maintains a physically contiguous finite length data space and that is updated when new data samples become available is referred to as a sliding window. After the processing is completed, the reference pointer is updated as follows:

$$RP_{new} = \begin{cases} RP_{old} + 1, & PH \leq RP_{old} < RT \\ PH, & IH = RT. \end{cases}$$

An output manager class has attributes holding output file name and the number of waveform events stored in the file. It also encapsulates methods for reading the waveform events stored in a result file and for storing waveform events detected by the waveform detection system in a result file according to the temporal order in which they occur.

3.3.1.5 Task Controller Class Objects

Two different classes of task controllers are involved to process the EEG collected from multiple derivations; they are a main task controller class object at the high level and many low level channel task controller class objects. The main task controller class object is a composite object that, as its parts, contains other objects such as a number of channel task controller class objects and input/output manager class objects. The main task controller class object instantiates the part objects, schedules their tasks, triggers the execution of the tasks according to the schedule, and collects the resultant output from them. By sending a message with necessary information to the part objects, the main task controller class object controls and communicates with them (the detailed explanation about how they interact is described in the next section "initial set-up and operation"). The channel task controller class object is also a composite object that has one or more waveform detector class objects as its parts. It instantiates the part objects, triggers their

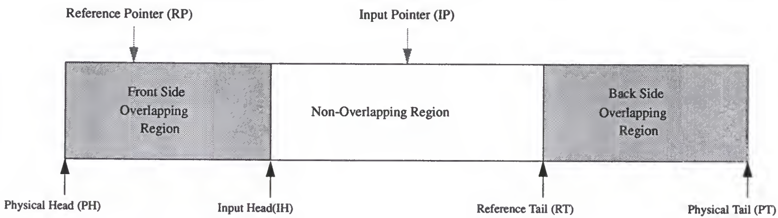


Figure 3-10. A scrollable buffer with two overlapping re-

waveform detection process, collects the resultant waveform events from its part waveform detector class objects, and sends them to the upper level main task controller class object.

The main task controller class has five attributes used for managing the lower level channel task controller and input/output manager class objects, which include the attributes, subject-name and number-of-channels. The other three attributes are used to link a main controller class object with its part objects. The procedural knowledge for controlling and scheduling the operation of its part objects is implemented as the encapsulated method "detect-events." The methods "purge-store" and "store-all" manage the storage of output results. The channel controller class has four attributes, channel-number, electrode-placements, number-of-detectors, and one attribute which serves as a thread connecting it to the part objects. Several methods are provided to retrieve and modify the attributes.

3.3.2 Initial System Set-Up and Operation

There are two ways of customizing a software system. A simple way is to modify the source code, but this requires the recompilation and linking of the modified code at a later time. We will refer to this approach as an early customization. Another approach is to provide a facility that can interpret a configuration file and customize a system according to the specification while the system is running. We will call this approach a late or run-time customization. Although the run-time customization is a relatively complicated in implementation, it does not require a source code modification and time-consuming recompilation process, which is repetitive, time-consuming and error-prone in nature. A run-time customization approach is taken here.

In the object-oriented approach, the run-time system configuration is accomplished when instantiating new objects. The description of an overall system configuration is normally presented in a form that the user can edit. Figure 3-11 shows the recursive instantiating of processing objects when the waveform detection system is initialized. The

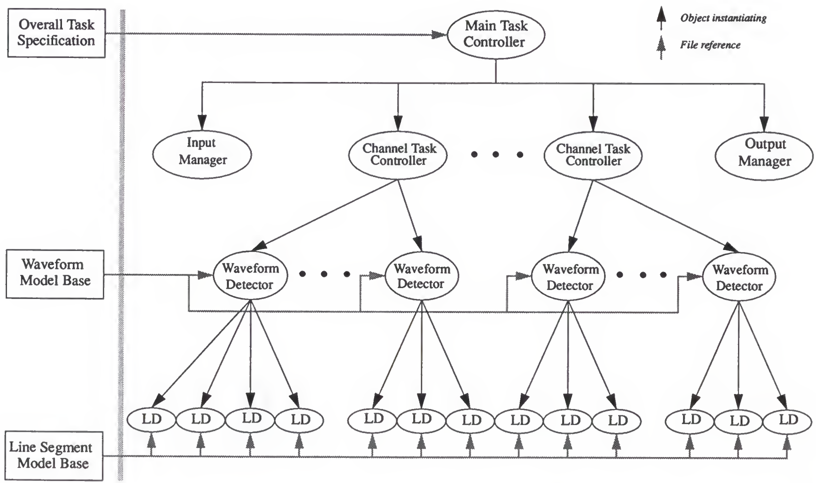


Figure 3-11. Recursive instantiating of the objects during the system initialization.

```
// Subject Name — r08gm11

Number-Of-Channels    :    8
Sampling-Rate         :    250
Number-Of-Databytes   :    1
Swapped-Bytes         :    0
50uV-Calibration      :    410

ChannelDescription
 Number               :    1
 Placements           :    Fp1 – Pz
 Detectors            :    <alpha><sigma><delta><eyemove><spike>
EndDescription

ChannelDescription
 Number               :    2
 Placements           :    F7 – Pz
 Detectors            :    <alpha><sigma><delta><eyemove><spike>
EndDescription

ChannelDescription
 Number               :    3
 Placements           :    T3 – Pz
 Detectors            :    <alpha><sigma><delta><eyemove><spike>
EndDescription

ChannelDescription
 Number               :    4
 Placements           :    T5 – Pz
 Detectors            :    <alpha><sigma><delta><eyemove><spike>
EndDescription

ChannelDescription
 Number               :    5
 Placements           :    F3 – Pz
 Detectors            :    <alpha><sigma><delta><eyemove><spike>
EndDescription

ChannelDescription
 Number               :    6
 Placements           :    C3 – Pz
 Detectors            :    <alpha><sigma><delta><eyemove><spike>
EndDescription

ChannelDescription
 Number               :    7
 Placements           :    P3 – Pz
 Detectors            :    <alpha><sigma><delta><eyemove><spike>
EndDescription

ChannelDescription
 Number               :    8
 Placements           :    O1 – Pz
 Detectors            :    <alpha><sigma><delta><eyemove><spike>
EndDescription
```

Figure 3-12. An example of the overall task specification.

// File Name — delta.wave

Waveform-Type	:	*Single-Phasic*
Waveform-Name	:	*delta*
Duration-Span-Ratio	:	*30%*
Amplitude-Span-Ratio	:	*40%*
Number-Of-Wave-Models	:	*2*

PatternDefinition

Waveform-Model-Name	:	*downward-delta*
Minimum-Total-Duration	:	*350 msec*
Maximum-Total-Duration	:	*2000 msec*
Average-Duty-Ratio	:	*65%*
Sequential-Segment-Pattern	:	*ncdl•pcdl*
Minimum-Pair-Period	:	*<350> (unit:msec)*
Maximum-Pair-Period	:	*<2000> (unit:msec)*
Minimum-Pair-Duty-Ratio	:	*<65> (unit:%)*

EndDefinition

PatternDefinition

Waveform-Model-Name	:	*upward-delta*
Minimum-Total-Duration	:	*350 msec*
Maximum-Total-Duration	:	*2000 msec*
Average-Duty-Ratio	:	*65%*
Sequential-Segment-Pattern	:	*pcdl•ncdl*
Minimum-Pair-Period	:	*<350> (unit:msec)*
Maximum-Pair-Period	:	*<2000> (unit:msec)*
Minimum-Pair-Duty-Ratio	:	*<65> (unit:%)*

EndDefinition

Figure 3-13. An example of waveform models (delta waveform model).

// Segment Name — pcdl

Upper-Contraction-Rate	:	*20%*
Lower-Contraction-Rate	:	*20%*
Minimum-Period	:	*112 msec*
Maximum-Period	:	*1000 msec*
Minimum-Amplitude	:	*50 uV*
Maximum-Amplitude	:	*300 uV*
Minimum-Slope	:	*0.2 uV/msec*
Maximum-Slope	:	*2 uV/msec*
Slope-Direction	:	*Positive*
Selection-Policy	:	*Max-Amplitude*

Note) ncdl is the same as pcdl except that the Slope-Direction is negative.*

Figure 3-14. Line segment model of the delta waveform model.

system initialization starts with reading an overall task specification file, of which names are usually constructed by concatenating the name of a subject under analysis to the following file extension ".conf."

The main task controller class object encapsulates methods for reading and interpreting the task specification file. As shown in Figure 3-12, the task specification file contains all the necessary information to launch the system initialization. The first four lines in the task specification file specify the binary data file format, while the rest part of the specification file is for the channel task controllers. In this example, the data file contains eight data channels and the system is configured to detect alpha, sigma, delta, eye-movement and spike waveforms from every EEG channel. Therefore, the main task controller class object instantiates eight channel task controller objects, each of which in turn instantiates waveform detector class objects for alpha, sigma, delta, eye-movement, and spike waveforms. The main task controller class object also instantiates the input and output manager class objects, and then they open data and result files, respectively. Eight scrollable data buffers are also created when the input manager class object is instantiated.

During the instantiation process, a waveform detector object reads the corresponding waveform model specification file which is named after the waveform name with the extension "wave," i.e., "delta.wave" is the specification file name of a delta waveform model. Figure 3-13 shows the delta waveform model file used in our experiments. According to the specification, the delta waveform consists of two different types of characteristic line segments: "ncdl" and "pcdl." The delta waveform detector object, therefore, instantiates two line segment detector objects. A line segment detector object is configured, according to the specification given by the line segment model specification file, which is named after the line segment type name with the extension".seg." Figure 3-14 shows a line segment model specification file of the line segment "pcdl."

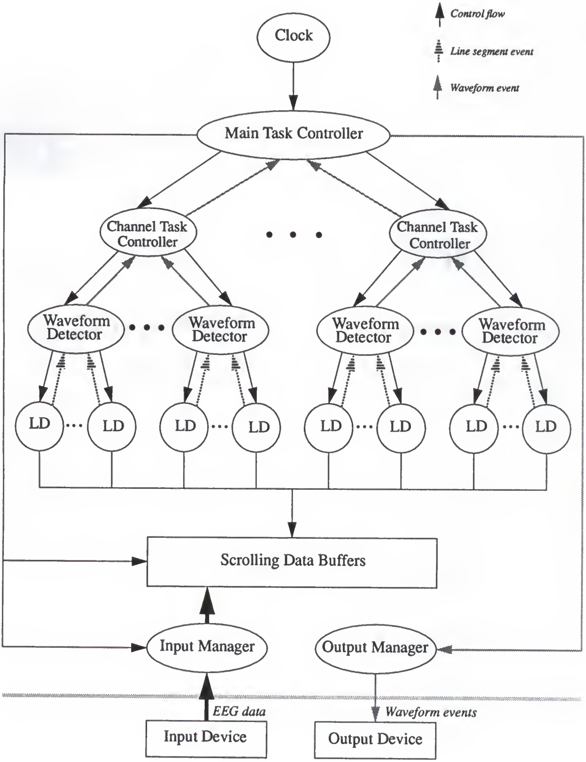


Figure 3-15. Control and data flow of the waveform detection system during operation.

As previously described, the waveform detection system presented here does not operate in a real-time situation, but it operates in a way similar to real-time waveform detection systems; the operation is centered on new data samples that have just become available. In order to mimic a real-time situation, time event messages, similar to the data ready signal of A/D converters, are generated by a clock object and sent to the main task controller class object. Figure 3-15 shows data and control flows during the waveform detection process.

Upon the receipt of the time event message, the main task controller object sends a data-read message to the input manager, and activates the channel task controller class objects according to the predefined order. Whenever the data-read message is received, the input manager object reads a frame of data from the input device, a file on a hard disk, and passes it to the scrolling buffers. In fact, the data are retrieved by blocks in order to speed up a hard disk access and stored in an input manager buffer until they are all transferred to the scrolling buffers frame by frame.

The activated channel task controller triggers the detection process of the lower level waveform detector objects, and then the waveform detector objects activate related line segment detector objects. A line segment event object is generated by the line segment detector objects whenever a representative line segment is successfully detected. The line segment event objects are transferred to the line segment queue of the upper level waveform detector object.

Every time a properly connected candidate line segment sequence that meets all requirements specified by the waveform model object is found, the waveform detector object generates a waveform event object and reports it to the upper channel task controller class object. Then the channel task controller class object sends the waveform event object to the main task controller class object. The waveform event object are sent to the output manager object and stored in a result file, according to the temporal order of their occurrences.

3.4 Experimental Results

The performance of the waveform detection system presented here entirely relies on the waveform models provided by the user. Thus, the selection of accurate waveform models is a preliminary but important step to improve the performance of the system. Initially, we set up a coarse waveform model by using the waveform characteristics generally admitted. The coarse waveform model was refined by carefully examining many waveform examples occurring in the EEG. The waveform morphology examination is performed in TDAT, in which the user can visualize EEG waveforms in detail and measure the amplitude and time information. Figure 3-16 shows the waveform examination process, using TDAT. Then the waveform model underwent a final refinement process, which involves a cycle of testing and tuning; the waveform detection system that employs the models processed a training data set, and the models were tuned to give good man/machine agreements. This cycle of refinement process was repeated until the man/machine agreements of the detection results satisfies a given threshold. Since it is performed by trial and error and needs human visual intervention, the refinement process is a time-consuming task.

All the sleep EEG records used in this study were first digitized with 10 bits, at a sampling rate of 500 Hz and were later stored on an optical disk, with 8 most significant bits at a sampling rate of 250 Hz, and they are all 1 hour long and include sleep stages 0, 1, 2, 3 and 4. Two sleep EEG records obtained from different subjects were used to refine the waveform models for alpha, sigma and delta activity. As shown in Figure 3-5(a) and (b), an alpha waveform is represented as a properly connected sequence of two alternating types of line segments: pcal and ncal. Figure 3-13 and Figure 3-14 specify the waveform and line segment models for delta waveform detection, and Figure 3-17(a) and (b) show the waveform line segment models for sigma waveform detection.

To evaluate the performance of an individual waveform detector by assessing the man/machine agreement is time-consuming and involves one or more EEGers. Thus, this

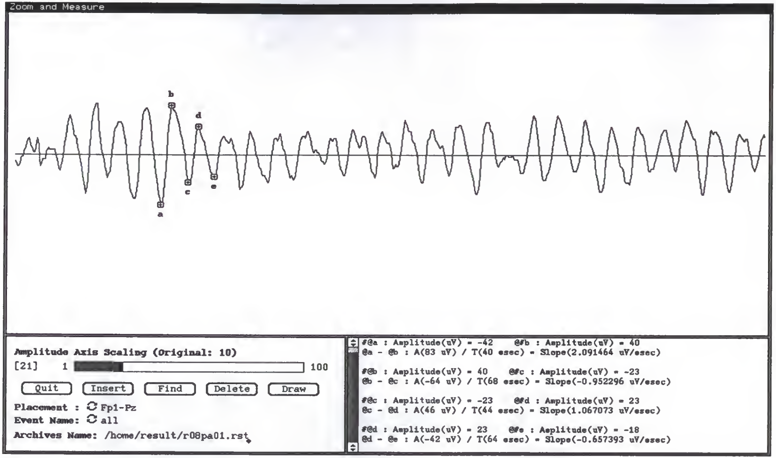


Figure 3-16. Scrutinization process of waveform morphology in TDAT.

```
// File Name — sigma.wave

Waveform-Type                 :    Multiple-Phasic
Waveform-Name                 :    sigma
Duration-Span-Ratio           :    40%
Amplitude-Span-Ratio          :    20%
Number-Of-Wave-Models         :    2

PatternDefinition
 Waveform-Model-Name          :    down-start-sigma
 Minimum-Total-Duration       :    188 msec
 Maximum-Total-Duration       :    292 msec
 Average-Duty-Ratio           :    70%
 Sequential-Segment-Pattern   :    ncsg•pcsg•ncsg•pcsg•ncsg•pcsg•ncsg
 Minimum-Pair-Period          :    <48><48><48><48><48><48> (unit:msec)
 Maximum-Pair-Period          :    <96><96><96><96><96><96> (unit:msec)
 Minimum-Pair-Duty-Ratio      :    <55><55><55><55><55><55> (unit:%)
EndDefinition

PatternDefinition
 Waveform-Model-Name          :    up-start-sigma
 Minimum-Total-Duration       :    188 msec
 Maximum-Total-Duration       :    292 msec
 Average-Duty-Ratio           :    70%
 Sequential-Segment-Pattern   :    pcsg•ncsg•pcsg•ncsg•pcsg•ncsg•pcsg
 Minimum-Pair-Period          :    <48><48><48><48><48><48> (unit:msec)
 Maximum-Pair-Period          :    <96><96><96><96><96><96>(unit:msec)
 Minimum-Pair-Duty-Ratio      :    <55><55><55><55><55><55><55> (unit:%)
EndDefinition
```

(a)

```
// Segment Name — pcsg

Upper-Contraction-Rate     :     8%
Lower-Contraction-Rate     :     8%
Minimum-Period             :     16 msec
Maximum-Period             :     40 msec
Minimum-Amplitude          :     15 uV
Maximum-Amplitude          :     100 uV
Minimum-Slope              :     0.4 uV/msec
Maximum-Slope              :     5.5 uV/msec
Slope-Direction            :     Positive
Selection-Policy           :     Max-Amplitude

Note*) ncsg is the same as pcsg except that the Slope-Direction is negative.
```

(b)

Figure 3-17. Sigma waveform specification. (a) Waveform model. (b). Line segment models.

evaluation uses a simple approach, i.e., a comparison with another system which has been evaluated with reference to human EEGer's visual scoring. The reference system, the waveform detector system developed by Smith, which reportedly achieved over 90% man/ machine agreement in detecting sleep spindles, was taken [Sm86]. For convenience, the waveform detector system developed by Smith will be referred to as the "Smith system" and the waveform detection system developed here will be referred to as the "Park system." This comparison procedure was carried out as follows: 1) both waveform detection systems independently process a test data set, 2) calculate the total time-duration of waveform detections occurring for every 30 second-long EEG epoch, 3) plot the total time-durations computed by both systems over every 30 second-long EEG epoch, and 4) compare the general trends.

The system performance test used three sleep EEG records collected from different subjects. The subjects whose EEG records were used for refining waveform models were excluded from the system performance test. Alpha, sigma and delta activities were detected, and the epoch-wise time-duration plots of three sleep EEG waveforms were obtained. Figure 3-18 through 3-20 display the epoch-wise time duration plots obtained from one of the sleep EEG records, where the black line denotes the detections by the Park system and the grey one denotes the detections by the Smith system. Figure 3-21 shows the sleep stages for every epoch scored by the Smith detector system in order to give a rough idea of the general waveform distribution trends.

In Figure 3-18, an abundance of alpha waveforms were detected by both systems in the first 30 epochs, which accord well with the corresponding sleep stage scoring. In later epochs, the two alpha distribution curves keep the same general trend, but more detections were made by the Park system. A scrutiny of each waveform detection over the epochs shows that the more frequent detections are mainly due to the use of relatively generous definitions of the alpha waveform in the Park system, i.e., the Park system requires four consecutive single waves, while the Smith system requires six consecutive single waves.

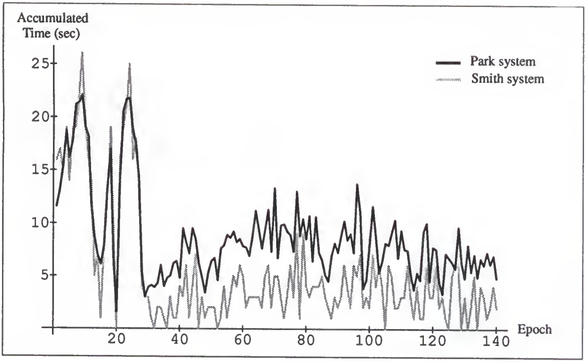


Figure 3-18. Comparison of alpha waveform detections.

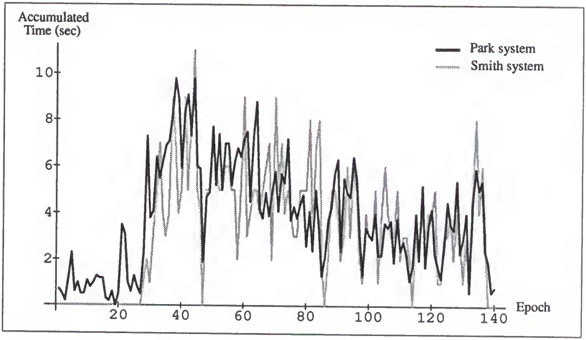


Figure 3-19. Comparison of sigma waveform detections.

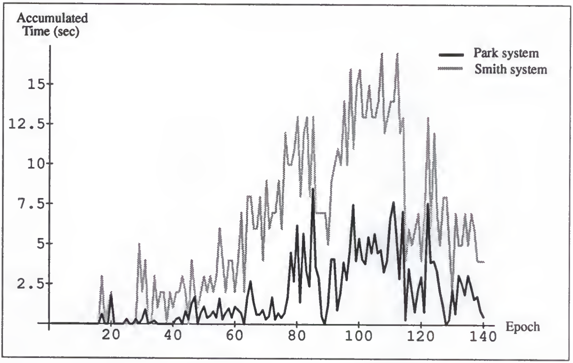


Figure 3-20. Comparison of delta waveform detections.

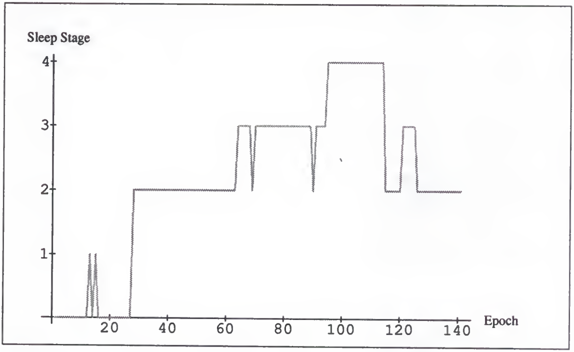


Figure 3-21. Sleep staging by the Smith system.

Thus, the Park system made more detections during the sleep stage 1, 2, 3, and 4 since they contains more marginal or ill-formed alpha waveforms. However, most of the alpha waveforms occurring during the sleep stage 0 were detected since it contains relatively well-formed alpha waveforms. For a similar reason, the Park system made some sigma waveform detections that the Smith system did not. Figure 3-21 shows more delta detections with the Smith system, but both curves follow the similar general distribution trends that conform to the given sleep stage scoring. Although the results obtained by the Smith system were not examined individually, it appears to perform well in the detection of the general trends of the EEG. This is mainly due to the preconditioning band-pass filter that manifests the primary component of a waveform of interest. The filter distorts the EEG to emphasize the activity similar to the waveform. Thus, the Smith system performs well in detecting the portion likely to have the waveform, especially in analyzing the general trends of the EEG, such as sleep stage scoring. On the contrary, the Park's system performs well in detecting the portions having a clear-cut waveform.

3.5 Discussion

The waveform detection system presented here contains a new search-oriented waveform detection algorithm, which performs the structural analysis of characteristic line segments obtained through a guided search of characteristic line segments. Known characteristics of EEG waveform morphology are used to limit the search space. The use of a search technique for detecting characteristic line segments requires a more elaborate structural analysis since adjacent line segments obtained from the search are not always connected together. However, the search technique brings in some useful properties. Even when a fast activity is superimposed over a slow activity, the algorithm can detect the fast activity without a priori signal conditioning, such as band-pass filtering used in the point-oriented algorithm, since it is based on the search for line segments satisfying predefined requirements. Thus, the filter design step can also be skipped when developing waveform

detectors that use the algorithm presented here. The unwanted waveform distortions due to the filtering can also be avoided.

The processing speed of this algorithm is dependent on the number and time durations of characteristic line segments being searched for. It is slower than the traditional point-oriented waveform detection algorithms using band-pass filtering. It took about 8 minutes to detect alpha waveforms from one- hour long, single channel EEG data, with Sun 3/50 under Unix operating system. However, the processing speed can be accelerated by using hierarchically arranged multiple processors, each of which is responsible for the search of a certain type, characteristic line segments. The algorithm is appropriate for parallel processing.

The algorithm is based on the observation that EEG waveforms can be readily distinguished by their characteristic line segments and structural features. This chapter also presented a unified waveform model specification format, in which the characteristic line segments and structural features of various EEG waveforms are specified by a set of numeric/symbolic parameters already defined.

Based on the search-oriented waveform detection algorithm and the unified waveform model specification format presented, a configurable waveform detection system was developed. The system is configurable in the sense that its operation entirely relies on the specifications provided by the user, without any modification of the source code; the overall system is configured according to the overall task specification, and the operation of each waveform detectors relies entirely on the corresponding waveform and line segment model specifications. Thus, different waveform detectors can be obtained by simply switching the input waveform and line segment model specification files.

In the experiments, the system gave results comparable with those obtained from one of the most reliable automated methods in the detection of three sleep EEG waveforms: alpha, sigma, and delta. The existence of the disagreements between the two systems are mainly due to the differences in the waveform definitions. Another reason for the

disagreements is that the Smith's system emphasizes the general trend of the EEG while the Park's system prefers a clear-cut waveform.

CHAPTER 4
KNOWLEDGE-BASED MODEL OF CONTEXTUAL ANALYSIS AND ITS APPLICATION TO EPILEPTIC SPIKE DETECTION

Most of automated EEG waveform detection systems that use only morphological features of waveforms as discriminators suffer from excessive false detections, mainly due to the insufficient discriminatory power of the morphological parameters they use. However, the EEGer can easily discriminate these false detections not only by looking at the morphological features of waveforms, but also by considering a wide EEG context, such as the existence of supporting and conflicting events, the state of consciousness of the subject and the electrode montage of EEG channels.

This chapter describes a distributed knowledge-based system model that uses a hypothesis-confirmation process for simulating the confirmatory and counteractive interactions of various contextual clues used in the EEGer's visual analysis. Representation schemes suitable for the contextual information and the EEGer's heuristic knowledge used in the human visual analysis are also discussed. Finally, a knowledge-based system is developed for the elimination of false positive detections generated by the spike waveform detector. The knowledge-based system employs the distributed knowledge-based system model and the representation schemes. The overall system that includes the spike waveform detector and the knowledge-based system for the elimination of false positive detections is tested by comparing its results with the epileptic spikes visually screened by two certified human EEGers.

4.1 Introduction

Several attempts have been made to use in automated EEG analysis the context clues used in the visual EEG analysis. Koffler and Gotman designed an algorithm that uses temporal and inter-channel relationships for recognizing groups of spikes and sharp waves and rejecting EMG artifacts [Ko85]. Glover et al. developed a knowledge-based system, which processes spatial and temporal context information available in sixteen channels of EEG, EKG, EMG, and EOG; the context information was used to eliminates false positives in the automated detection of epileptic sharp transients in the EEG [Gl89]. Jansen and Dwant developed a frame-based expert system in which multichannel synchrony information was taken into account to reliably detect sleep spindles and K-complexes [Ja89].

The aim of this study is to provide a more systematic way of representing various spatial and temporal EEG context and of simulating the confirmatory and counteractive interactions between various contextual clues in the EEGer's visual analysis. Our observation of the EEGer's visual EEG analysis is as follows: (1) the EEGer first looks for a main event (e.g., a spike in detecting epileptic EEG activity) when he visually scores multichannel EEG data; (2) when an EEG segment containing the main event is found, the EEGer establishes an initial hypothesis (e.g., the hypothesis that the spike is epileptogenic); (3) the EEGer looks for confirmatory and counteractive contextual clues which are distributed either spatially or temporally; (4) taking into account the contextual clues, the EEGer finally confirms or rejects the initial hypothesis, i.e., when there exist supporting contextual clues enough for the confirmation of the initial hypothesis, the initial hypothesis will be confirmed (e.g., the spike is epileptogenic), however the initial hypothesis will be rejected when there exist only conflicting contextual clues (e.g., the spike is a false positive detection due to sharp but normal activities). When both supporting and conflicting contextual clues co-exist, the EEGer may resolve the conflict according to his own criteria, or postpone the determination of the official view until a new evidence that affects the

decision is available. This process of setting up an initial hypothesis and confirming it through the use of contextual clues will be called a "hypothesis-confirmation process."

In this study, a distributed knowledge-based system model is presented to efficiently simulate the confirmatory and counteractive interactions between various contextual clues during the hypothesis-confirmation process. In the distributed knowledge-based system model, one Coordinating Rule-Based System (CRBS) supervises many Specialized Rule-Based Systems (SRBS) that are simultaneously cooperating and competing. Each SRBS is responsible for a particular type of contextual clues, thus it maintains a relatively small amount of knowledge needed only for the specialty. The distribution of heuristic knowledge for EEG contextual analysis in many expert systems improves the processing speed and raises the software maintainability and modularity. Knowledge representation schemes suitable for various different types of spatial and temporal context information and heuristic knowledge used in the visual EEG analysis are also discussed.

Based on the distributed knowledge-based system model and the knowledge representation schemes, a knowledge-based contextual analysis system is developed for screening out false positive detections, which are generated by the spike waveform detector. The problem of detecting epileptic spikes is a relatively complex task even to human EEGers. It involves a variety of spatial and temporal context information, such as the existence of subsequent slow waveforms, the spatial and temporal distributions of sharp transients, the amplitude and frequency asymmetry between different hemispheres, the state of consciousness of the subject and the type of the montage utilized.

4.2 Distributed Knowledge-Based System Model for Contextual Analysis

The distributed knowledge-based system model presented here comprises two different types of rule-based systems: one Coordinating Rule-Based System (CRBS) and many Specialized Rule-based Systems (SRBS), which are organized as shown in Figure 4-1. The coordinating rule-based system supervises the overall hypothesis-confirmation

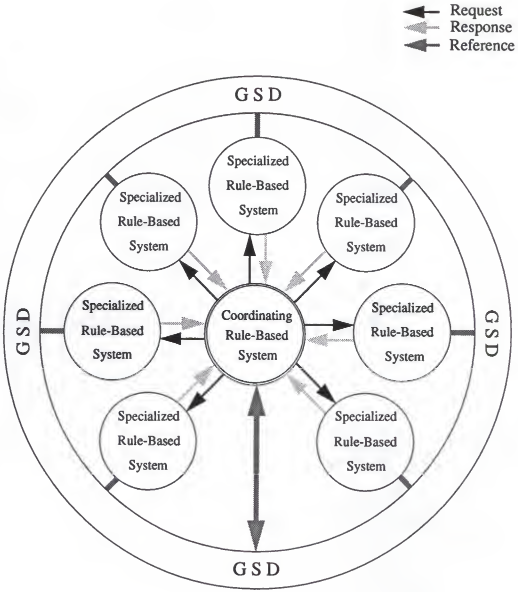


Figure 4-1. The organization of a coordinating rule-based system and many specialized rule-based systems.

process performed by the distributed knowledge-based system model; it formulates an initial hypothesis, determines what kinds of contextual clues are required, activates the corresponding specialized rule-based systems to get the contextual clues, and finally integrates them to determine whether to confirm or to reject the initial hypothesis. Every specialized rule-based system takes charge of a particular type of contextual clues.

4.2.1 Specialized Rule-Based System (SRBS)

A specialized rule-based system is identified by the type of the contextual clues it deals with. The contextual clues used in human visual EEG analysis can be classified into four different categories: main, veto, supporting, and conflicting contextual clues. The initial hypothesis is confirmed when a main contextual clue is obvious, and the existence of any veto contextual clue may result in the invalidation of the initial hypothesis. Supporting contextual clues help to confirm the initial hypothesis, while conflicting contextual clues weaken the validity of the initial hypothesis.

According to the type of the contextual clues they deal with, SRBSes are also divided into four categories: main, veto, supporting, and conflicting SRBSes. As a result of the analysis, every SRBS gives the upper level CRBS an information about whether the contextual clue it deals with exists or not, which will be called the view of the SRBS. When the contextual clue is obvious or absent, the view is conclusive. However, when the contextual clue is not obvious based on the evidences currently available, the view is nonconclusive. The conclusive view of the main SRBS or any veto SRBS terminates the hypothesis-confirmation process, while supporting and conflicting SRBSes provide contextual clues when a decision cannot be made by the main and veto SRBSes.

Every SRBS consists of four components: a task controller, an inference engine, a knowledge base, and a scene interpreter (see Figure 4-2). The task controller is responsible for scheduling all tasks carried out by the SRBS; it first activates the scene interpreter when initiated by the CRBS, and then it triggers the reasoning process of the inference engine.

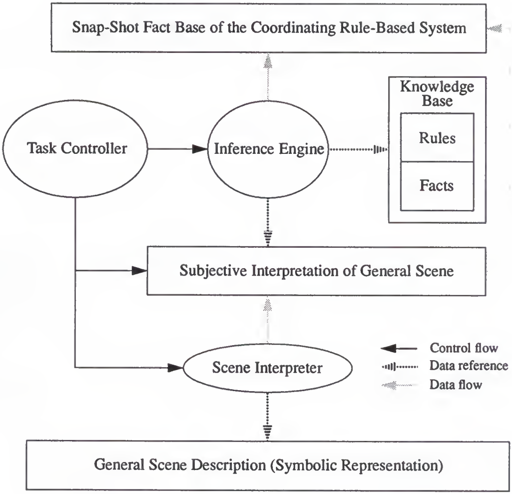


Figure 4-2. The structure of a specialized rule-based system.

During the reasoning process, the inference engine tests two inference goals, a main goal and a counter goal. The counter goal is first tried, and if it is achieved, the SRBS will give the CRBS the view that the corresponding contextual clue does not exist. If the counter goal cannot be achieved, then the main goal is tried. When the main goal is successfully achieved, the SRBS will inform the CRBS as its official view that the corresponding contextual clue exists. When both the goals are failed, the SRBS will give a nonconclusive view to the CRBS, which is of no importance in the hypothesis-confirmation process.

The knowledge base consisting of domain rules and facts determines the perspective of the SRBS, together with the scene interpreter. General scene description (GSD) of Figure 4-2 denotes a stream of events occurring in the EEG segment currently being analyzed. Every SRBS interprets the GSD from a unique point of view. The subjective scene interpretation stores the subjective description of the GSD obtained by the scene interpreter.

4.2.2 Coordinating Rule-Based System (CRBS)

A coordinating rule-based system formulates an initial hypothesis, and supervises the overall hypothesis-confirmation process. If the main SRBS produces a conclusive view, the CRBS will terminate the hypothesis-confirmation process and declare the view of the main SRBS as an official view of the CRBS. On the other hand, if there is any veto SRBS having a conclusive view, the CRBS will reject the initial hypothesis and terminate the hypothesis-confirmation process at that point. If both main and veto SRBSes have no conclusive view, the CRBS requests contextual clues to the supporting and conflicting SRBSes. The CRBS combines the contextual clues, and if there exists any conflict between them, resolves it, and finally determines the official view.

The coordinating rule-based system consists of six components: an agenda, a task controller, an inference engine, a rule base, a permanent fact base, and a snap-shot fact base (see Figure 4-3). The agenda keeps the initial hypothesis, and specifies what kinds of tasks

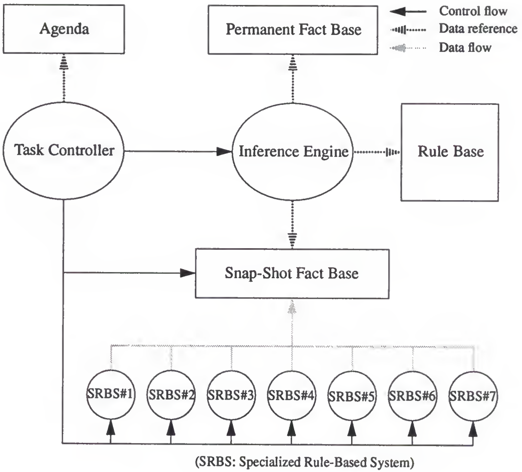


Figure 4-3. The structure of a coordinating rule-based system.

are to be done during the hypothesis-confirmation process. The task controller requests appropriate SRBSes to execute the tasks specified by the agenda and to give their views. The snap-shot fact base stores the views of the SRBSes on the current scene. The permanent fact base contains time-invariant facts, such as spatial relationships of electrode placements in the 10-20 International System. The rule base contains the rules used for integrating the contextual clues and resolving any conflict between them. The inference engine determines an official view by using the rules and facts, when both main SRBS and veto SRBSes have no conclusive view. The inference engine has two inference goals: a main goal and a counter goal. In the reasoning process, the inference engine first attempts to achieve the main goal, while the counter goal is attempted, only when the main goal is not achieved.

4.2.3 Organization and Control Strategy

A CRBS and various SRBSes are hierarchically organized as shown in Figure 4-4. A CRBS is located at the high level to supervise the low-level SRBSes. Since the SRBSes are limited to communicate only with the high-level CRBS, all the interactions between SRBSes are accomplished through the CRBS.

The CRBS supervises the hypothesis-confirmation process; it establishes an initial hypothesis when a specific type of events are detected at the GSD (the specific type of events are called main events), and then initiates the hypothesis-confirmation process. The CRBS first examines veto SRBSes whether they have contextual clues enough to veto the initial hypothesis. Any veto of the veto SRBSes will reject the initial hypothesis and terminate the hypothesis-confirmation process of the CRBS. When there is no veto, the CRBS asks the main SRBS to give its view.

The conclusive view of the main SRBS will cause the termination of the current hypothesis-confirmation process; if the main goal of the main SRBS is achieved, the process is terminated with the initial hypothesis successfully confirmed, while the fulfillment of the counter goal lead to the rejection of the initial hypothesis. Unless the view

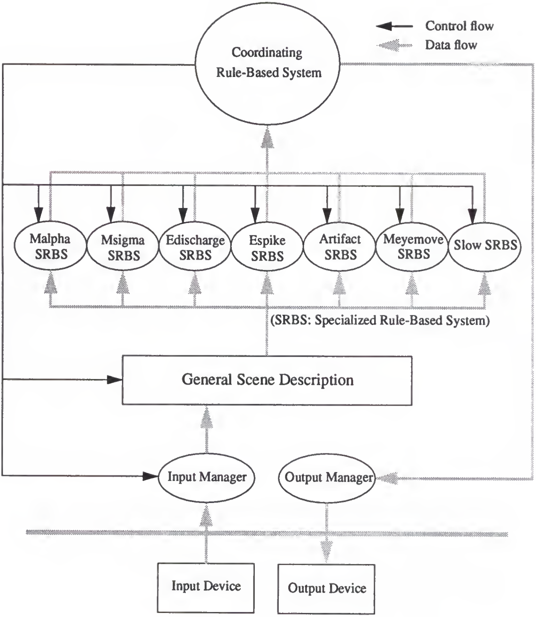


Figure 4-4. The organization of objects in the knowledge-based contextual analysis system for eliminating false detections in the detection of epileptic spikes.

of the main SRBS is conclusive, the CRBS activates supporting and conflicting SRBSes to collect contextual clues. The views of supporting SRBSes are referred to as "supporting views," while the views of conflicting SRBS as "conflicting views." The CRBS stores all the views from supporting and conflicting SRBSes in its snap-shot fact base, and triggers the reasoning process of the inference engine.

Generally, if the snap-shot fact base contains supporting views strong enough to confirm the initial hypothesis, the main goal is achieved in the reasoning process, while the reasoning process with only conflicting views tends to achieve the counter goal. If the supporting and conflicting views are mixed, the reasoning process will spend more time in resolving the inconsistency. If the CRBS fails to resolve the inconsistency, the determination of an official view will be postponed, until new view becomes available.

4.3 Representation Schemes for Contextual Information and Heuristics

The visual EEG analysis involves various types of contextual knowledge, which are broadly divided into two classes: spatial context information and temporal context information. Temporal relationships based on time intervals are used to represent temporal EEG context. The 10-20 International System of electrode placement is represented by using an object-oriented approach to describe spatial EEG context. The heuristics the EEGer uses in the visual analysis are encoded into production rules.

4.3.1 Contextual Information Representation

To represent temporal relationships of EEG events occurring either in the same or different channels, some definitions are adopted from James Allen's Interval Calculus. In the interval calculus, a time interval is represented as a pair of end-points over some arbitrary, linearly, ordered domain; the temporal relationships between time intervals are based on thirteen binary relationships. Seven of the basic temporal relationships are illustrated in Table 4-1. Six other relationships are defined as the inverses of the basic

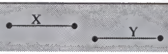
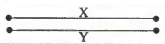
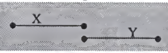
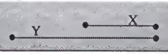
Relation	Pictorial Example
X before Y	
X equal Y	
X meets Y	
X overlaps Y	
X during Y	
X starts Y	
X finishes Y	

Table 4-1. Seven basic temporal relationships.

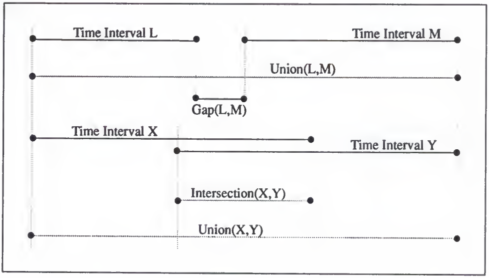


Figure 4-5. Examples of Union, Intersection and Gap operations.

temporal relationships; notice that the temporal relationship “equal” has no inverse temporal relationship.

Three binary temporal operators, “Union,” “Intersection,” and “Gap,” are defined to generate a new time interval from two time interval arguments. The temporal operator “Union” returns the shortest time interval that includes the two time interval arguments, while the temporal operator “Intersection” returns the longest time interval that is common to the two time interval arguments. The temporal operator “Gap” returns the longest time interval between the two time interval arguments. If there is no time between the two time interval arguments, the Gap operator returns NULL. Figure 4-5 illustrates the operation of the three temporal operators. The unary operator “Duration” is used to compute the metric index of a time index. Given the time interval L = (a, b), where a is always less than b, Duration(L) returns (b-a).

The description of the spatial distribution of EEG events requires the relative spatial relationships of electrode placements. In this study, the 10-20 International System (shown in Figure 4-6) is encoded by using an object-oriented approach. The encoded knowledge of the electrode system is mainly used to check the adjacency of EEG events obtained from multichannel EEG data. In the object-oriented approach, every electrode placement is modeled as an object with eleven attributes. Two attributes, “brain-area” and “hemisphere,” are used to identify the underlying brain areas and the side of hemisphere which the electrode is located at, respectively. The remaining nine attributes are “front-just, “front-right,” “right-just,” “right-rear,” “rear-just,” “rear-left,” “left-just,” “left-front,” and “symmetric” describe the relative position in the 10-20 International System, each attribute keeps the pointer indicating the electrode placement object located in the corresponding direction. Figure 4-7 illustrates nine directions at the electrode location C3, which are used as the attributes. The composite object “electrode-system,” which has 21 electrode placement objects as its parts, provides an access point to these electrode placement objects together with some methods for checking the adjacency of electrode placements.

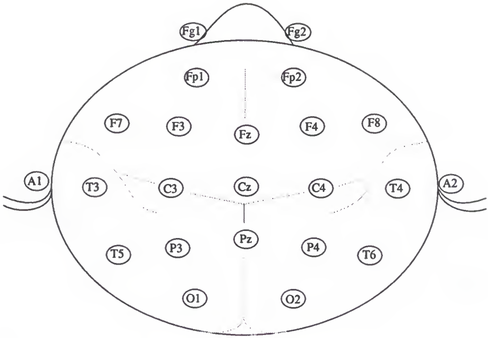


Figure 4-6. The 10-20 International System of electrode placement.

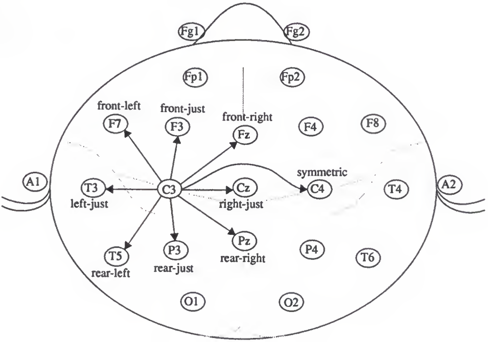


Figure 4-7. Nine directions used for representing the relative location of C3 in the 10-20 International System.

The particular arrangement of different derivations from which the EEG is displayed simultaneously in a paper tracing is termed a montage. There are two basic types of EEG montage: referential and bipolar. The principle of referential montages involves measuring the electrical activity at different electrodes simultaneously, in comparison with a common reference electrode. In bipolar montages, the potential difference between two electrodes placed on the scalp is displayed. Unlike the case of referential recording, both electrodes are considered to be active, and the varying difference in voltage between the two is recorded.

Since each channel displays the EEG from a pair of electrode placements (a derivation), two pairs of electrode placements must be taken into account to check the adjacency of two EEG events from different channels. However, since referential montages use a single electrode placement as a common reference, only two active electrode placements are needed to check the adjacency of EEG events occurring in the EEG using the referential montage. For example, an event occurring in the derivation C3 - Pz is said to be adjacent to one in the derivation F3 - Pz, since they use the same reference Pz, and C3 and F3 are adjacent. All the EEG data used in this study, are recorded by using referential montages, and thus the adjacency relationships of EEG events can be inferred directly from the encoded spatial knowledge of the electrode-system object and the electrode placement objects.

An EEGer obtains most of her/his contextual information from the EEG that can be seen in a single glance, which will be termed a "scene." 20 second-long multichannel EEG data are used as the scene, which is described by a temporally-ordered list of EEG events. More definitions are introduced to describe spatio-temporal and clinical relationships of the EEG events. The spatio-temporal and clinical relations used for representing the EEG context are shown in Figure 4-8.

Overlapping EEG events in a certain time period are "contending" if they are occurring in the same channel. The overlapping events in different channels are

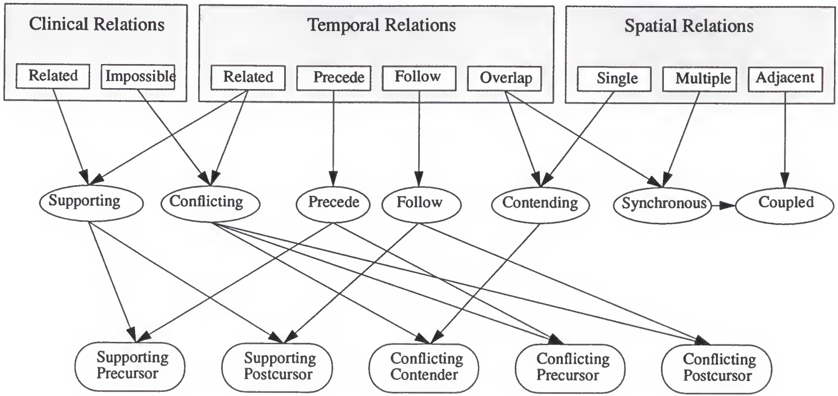


Figure 4-8. Spatio-temporal and clinical relations used for representing the EEG context.

"synchronous." Synchronous events which occur in adjacent channels are "coupled." An event having three or more coupled, same-type events has a strong spatial-support, while the one having two coupled, same-type events has a normal spatial-support. An event having only one coupled, same-type event has a weak spatial-support, and an event that has no coupled, same-type event has poor spatial-support.

Events occurring in the temporal proximity are "supporting" if they are consistent in clinical context, while the events of no clinical causality are "conflicting." If an event occurs before another, the former "precedes" the latter, and the inverse of "precedes" is "follows." The temporal operator "Gap" is used to check whether an event can influence another event. Let T_1 be the time interval of an event, and T_2 be that of another event. If Duration(Gap(T_1, T_2)) or Duration(Gap(T_2, T_1)) is less than a given threshold value, they are temporally "related."

If two temporally-related events occurring in the same channel are supporting, the preceding event is called a "supporting-precursor" of the following one. Conversely, the following event is called a "supporting-postcursor" of the preceding one. If the two events are conflicting, they are called "conflicting-precursor" and "conflicting-postcursor" of the other, respectively. If two events are both contending and conflicting, they are called a "conflicting-contender" of the other. No event can be both contending and supporting. The followings is an example of the context representation using the definitions described above.

EEG scene: Channel 1 has a spike overlapping with an overlapping eye-movement and followed by a slow wave. The spike and the slow wave are occurring in a close temporal proximity. Three other channels adjacent to the channel 1 also have spikes occurring simultaneously with the spike. Clinically, an epileptic spike has a tendency to accompany a slow wave, but there is little probability that epileptic spikes occur simultaneously with eye-movements.

Here, the context of the scene of which the reference is the spike occurring in the channel 1 is represented as follows ("this" denotes a reference):

Context representation:

(is this spike)

(contending this eye-movement)

(precede this slow-wave) or (follow slow-wave this)

(temporally-related this slow-wave) or (temporally-related slow-wave this)

(spatial-support this strong)

(supporting slow-wave this) or (supporting this slow-wave)

(conflicting this eye-movement)

The above representation can be more simplified as follows:

(is this spike)

(has this conflicting-contender)

(has this supporting-postcursor)

(spatial-support this strong)

4.3.2 Heuristic Knowledge Representation

When analyzing spatial and temporal context information collected from multichannel EEG data, the EEGer uses the heuristic knowledge obtained from his clinical experience and training. Such heuristic knowledge mainly consists of many fragmented facts, which can hardly be described by a concise, unified theory. Production rules are known to be an appropriate knowledge representation paradigm for such heuristic knowledge [Ba89]. In this study, production rules are used to represent the heuristic knowledge. The definition of rules consists of a name, a condition part (or left-hand side), and an action part (or right-hand side). A rule is declared as follows:

RULE-NAME <name>

IF <condition-part>

THEN <action-part>

END-RULE

A rule declaration always begins with the keyword "RULE-NAME," which is followed by the name, and it ends with the keyword "END-RULE." No space is allowed in the ASCII character rule name. The condition part consists of one or more conditions; each condition is represented as a list, and adjacent conditions are assumed to be linked by implied AND boolean operators. A condition has the form (predicate term term ...), where the term is either a constant or a variable, and the predicate describes the spatial or temporal relationship of the following terms. The action part comprises only one action list. The following is an example of contextual analysis rules for detecting epileptic spikes.

```
RULE-NAME      Epileptic-Spike-Detection-Rule1
   IF          ((spatial-support ?x strong)(has ?x supporting-postcursor)
                (has-no ?x conflicting-contender))
   THEN        (is ?x espike)
END-RULE
```

In the above rule declaration, ?x denotes a variable, of which occurrences in the condition and action lists are bounded to the same constant at a time during the reasoning process.

4.4 Knowledge-Based Contextual Analysis System for eliminating False Positives in the Detection of Epileptic Spikes

In order to be used in clinical practice, an automated detection system for epileptic spikes reliably detects epileptic spikes without generating too many false detections. Several attempts have been made to eliminate the false detections generated by the spike detectors that use only the morphological features [Gl89, Ko88, Ol83]. Context information, involving temporal as well as spatial (across all signal channels) analysis, is used to eliminate spike-like activity in the EEG which may be artifact or sharp but normal

activity. However, most of the methods still generate a large number of false positive detections [Kt89].

In this section a distributed knowledge-based system model and different knowledge representation schemes described above are used to build a knowledge-based contextual analysis system for eliminating false detections generated by the spike waveform detector that uses only the morphological features of epileptic spikes. The knowledge-based contextual analysis system presented here uses a hypothesis-confirmation process to analyze more contextual clues in a way similar to the EEGer's visual analysis. The contextual clues, including the temporal and spatial distribution of spikes, the existence of supporting and conflicting evidences occurring during the time periods preceding and following the spikes, were collected from multiple-channel EEG data by using the waveform detection system described in the chapter 3.

4.4.1 System Organization

The knowledge-based contextual analysis system for eliminating the false detections consists of a global scene description, an input manager, an output manager, a Coordinating Rule-Based System (CRBS), six specialized rule-based systems (SRBS), which are espike, edischarge, slow-wave, malpha, msigma, artifact, and meye-move SRBS. Figure 4-4 shows the organization of the knowledge-based contextual analysis system, with displaying control and data flow between the components.

The general scene description (GSD) is a doubly-linked list of waveform events, which were detected by the waveform detection system described in the chapter 3 (the configuration of the waveform detection system is shown in Figure 4-9). The GSD contains all the EEG waveform events occurring in a relatively short epoch, which will be called an EEG scene. It corresponds to "the amount of EEG data the EEGer usually looks at in a single glance." As the analysis proceeds, the GSD is being updated; it accepts new waveform events from the input manager and discards outdated waveform events, i.e., the

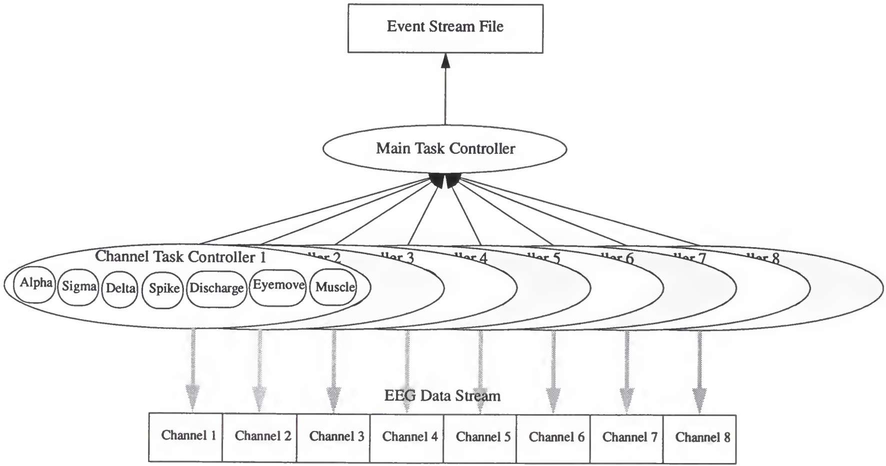


Figure 4-9. The configuration of various waveform detectors for automated epileptic spike detection.

GSD is assumed to be a symbolic representation of the EEG scene currently being analyzed.

As the time is updated, the input manager reads the waveform events from the input device (currently, a data file on a hard disk) according to the temporal order and passes them to the GSD, while the output manager sends the waveform events validated in the hypothesis-confirmation process to the output device (currently, a data file on a hard disk).

The CRBS monitors the GSD to look for a spike waveform event, and if it finds a spike waveform event, the CRBS establishes an initial hypothesis to initiate the hypothesis-confirmation process. However, the hypothesis-confirmation process is hibernated until enough spatial and temporal context information becomes available. The CRBS activates the low-level SRBSes by sending a message specifying the time interval and the channel on which the hypothesis-confirmation process is focused, which will be termed a focus interval and a focus channel, respectively. The CRBS first asks the artifact SRBS, which is a veto SRBS, whether it will give a veto or not. The veto of the artifact SRBS will terminate the hypothesis-confirmation process with invalidating the initial hypothesis. If there is no veto, then the CRBS continues the hypothesis-confirmation process with requesting the view of the espike SRBS, which is a main SRBS. Upon the request, the espike SRBS analyzes the spatial distribution of spikes and the temporal distribution of contextual clues in the focus channel. If there exist supporting or conflicting evidences enough either to confirm or to reject the initial hypothesis, the espike SRBS will post a corresponding conclusive view on the Snap-Shot Fact Base and terminate the hypothesis-confirmation process with validating the initial hypothesis.

When the espike SRBS cannot give the conclusive view, it informs the CRBS of the detailed description of the current situation instead of posting its view, and the CRBS requests the views of supporting and conflicting SRBSes. The malpha, msigma, meyemove, and artifact SRBSes provide the CRBS with conflicting evidences obtained from multichannel EEG data, while the edischarge and the slow-wave SRBSes provide

with supporting evidences obtained from multichannel EEG data. The SRBSes analyze multichannel EEG data to determine their views on the initial hypothesis, and post them on the Snap-Shot Fact Base of the CRBS. Taking into account the supporting and conflicting evidences, the CRBS finally determines whether the initial hypothesis is valid or not.

4.4.2 System Initialization

The knowledge-based system developed here consists of hierarchically-organized objects. The run-time customization (or late customization) approach is used to configure the overall system. Here, the configuration of a system is determined by three types of external specification files, which are provided by the user at run-time; they are overall task specification, specialized task specification and input/output specification files.

The overall task specification file (shown in Figure 4-10) specifies the types of tasks needed to validate epileptic spikes (for notational simplicity, the epileptic spikes is termed the "espike"). The specification file of Figure 4-10 shows that the detection process of the epileptic spikes involves eight SRBSes: a main, a veto, four conflicting, and two supporting SRBSes. As describe above, the espike SRBS is a main SRBS, and the artifact SRBS is a veto SRBS. The malpha, the msigma, the meyemove, and the artifact SRBSes are four conflicting SRBSes, while the slow-wave and the edischarge SRBSes are supporting SRBSes.

Together with the focus interval, the two parameters, Detect-Latency-Period and Queue-Lingering-Period, specify the time interval of the current EEG scene, which is depicted by the GSD. The GSD stores all the waveform events occurring in the time interval between (t_{start} - Queue-Lingering-Period) and (t_{end} + Detect-Latency-Period), where t_{start} and t_{end} indicate the start-time and end-time of the focus interval. The two list parameters, Inference-Goal and Counter-Goal, specify the goals of the reasoning process performed by the CRBS. The rules needed for the reasoning process are inserted between

```
// File name — espike.mcm

// Main Specialized Rule-Based System
Main-Event                  :        espike
Min-Overlap-Duration        :        30 msec

// Veto Specialized Rule-Based Systems
Veto-Events                 :        <artifact>

// Contending Specialized Rule-Based Systems
Conflicting-Events          :        <malpha><msigma><meyemovement><artifact>

// Supporting Specialized Rule-Based Systems
Supporting-Events           :        <edischarge><slow>
Supporting-Intervals        :        <250><1200> (unit:msec)

// Parameters for General Scene Description
Detect-Latency-Period       :        1000 msec
Queue-Lingering-Period      :        1000 msec

// Parameters for Focus Estimator
Focus-Constitute-Events     :        2 (synchronous events)

// Parameters for Scrutinizing Process (or Reprocessing)
Scrutiny-Policy             :        Moderate
Scrutiny-Synch-Events       :        2 (synchronous events)

// Goals of the Coordinating Rule-Based System
Inference-Goal              :        (is this espike)
Counter-Goal                :        (is-not this espike)

// Rules for the Coordinating Rule-Based System
Domain-Rule-Specification

 (Contextual analysis rules are located here)

End-Rule-Specification
```

Figure 4-10. Overall task specification file (file name: espike.mcm).

the keywords "Domain-Rule-Specification" and "End-Rule-Specification" in the bottom of the figure.

During the instantiation, every SRBS reads the system parameters from the corresponding specialized task specification file, which determines the particular behavior of the SRBS. Figure 4.11 shows the specialized task specification file of the main espike SRBS. The waveform event "spike" is used as the main event, on which the operation of the SRBS is centered. Muscle, alpha, or sigma waveform events contending with the main waveform event "spike" are conflicting contenders, while slow-wave and discharge waveform events following the main waveform event provide supporting evidences when they occur within either Precursor-Zone or Postcursor-Zone. The Precursor-Zone-Length and Postcursor-Zone-Length are used to specify the Precursor-Zone and the Postcursor-Zone as illustrated in Figure 4-12. The Min-Overlap-Duration specifies the minimum time duration in which synchronous spike waveform events occurring in different channels should overlap (see Figure 4-13 (a)).

Every SRBS has main and counter goals for its inference process. The inference process first starts with the counter goal, and if it is achieved, then the SRBS will give the CRBS the view that is conclusive but opposed to what the SRBS originally intends to claim. For example, the main SRBS will give the view that the initial hypothesis is invalid, and the veto SRBS won't put a veto, and the supporting or conflicting SRBSes will claim that there is no supporting or conflicting evidence.

Only when the counter goal fails to be achieved, the main goal is tried, and if it is achieved, the SRBS will give the conclusive view which it originally intends to claim. For example, the main SRBS will confirm the initial hypothesis, and the veto SRBS will put a veto. The supporting SRBSes will provide supporting evidences, while the conflicting SRBSes will provide conflicting evidences. The specialized task specification files for malpha, msigma, artifact, meyemove, edischarge and slow-wave SRBSes are shown in APPENDIX B.

```
// File name — espike.mcd

// Main Event and Constituent Waveform
Main-Event-Name              :       espike
Constituent-Waveform         :       spike
Min-Overlap-Duration         :       30 msec

// Enumeration of Main Event Models
Event-Model-Specification
      Model-Name                   :       espike1
      Supporting-Precursors        :       None
      Precursor-Zone-Length        :       None
      Supporting-Postcursor        :       <delta><discharge>
      Postcursor-Zone-Length       :       <1000><250> (unit:msec)
      Conflicting-Waveforms        :       <muscle><alpha><sigma>
      Conflicting-Overlap-Duration:        <30><30><30> (unit:msec)
End-Model-Specification

// Goals of the Specialized Rule-Based System
Inference-Goal               :       (is this espike)
Counter-Goal                 :       (is-not this espike)

// Rules for the Specialized Rule-Based System
Domain-Rule-Specification

 (Multichannel contextual analysis rules for espike are located here)

End-Rule-Specification
```

Figure 4-10. Specialized task specification file (file name: espike.mcd).

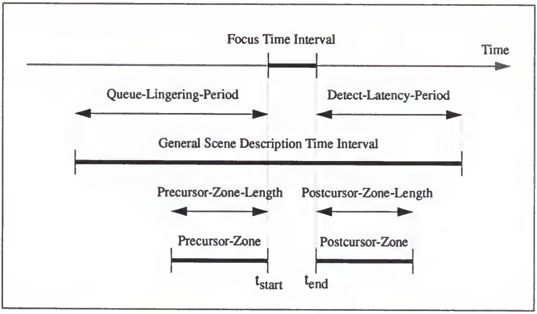


Figure 4-12. The time interval of General Scene Description and Precursor-Zone and Postcursor-Zone.

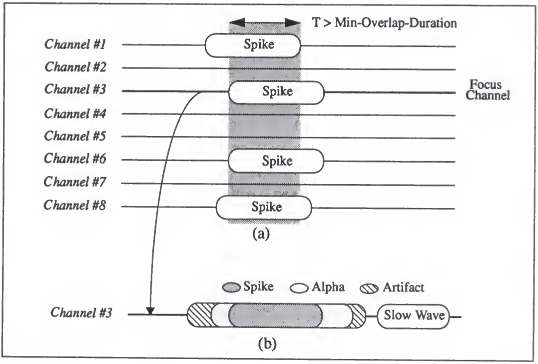


Figure 4-13. The subjective scene alignments. (a) spatial scene (b) temporal scene.

As described above, the overall task specification file specifies how many and what kinds of SRBSes are required. It also contains some parameters for controlling the overall hypothesis-confirmation process. The configuration of the overall system is determined by the overall task specification file during the system initialization stage. If there exists any complex object having component objects that need to be instantiated, the components objects will also be instantiated. This recursive object instantiation during the system initialization stage is shown in Figure 4-14.

The overall task specification file is named after the main event being analyzed and with the extension "mcm," which stands for "multiple channel mediation." Since the goal of the system developed here is to discriminate epileptic spikes, which is termed "espike," the name of the overall task specification file is "espike.mcm." The specialized task specification files are named after the event the corresponding SRBS is mainly interested in and with the extension "mcd," which stands for "multiple channel detection." The main SRBS which is interested in muscle artifacts uses the specified task specification file "artifact.mcd."

4.4.3 General Scene Description

For the analysis of spatial and temporal context information, the EEG scene is first represented as a stream of waveform events that are related to the detection of epileptic spikes. The waveform detection system described in the chapter 3 was used to detect the waveform events from the multiple channel EEG data. Figure 4-9 shows the configuration of the waveform detection system, which detects spikes, alpha waves, sigma waves, slow-waves, eye-movements, and muscle activities from 8 channel EEG data. The waveform events detected are stored in a file on a hard disk, and later retrieved for the contextual analysis.

The EEG scene was defined as the amount of EEG data which contains most of the contextual clues human EEGers look at in the visual EEG analysis. The general scene

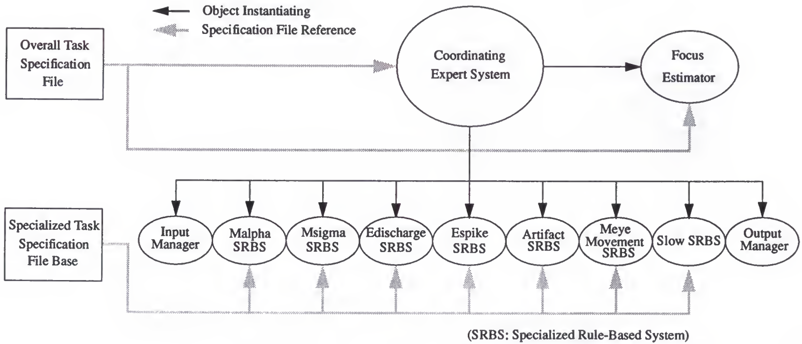


Figure 4-14. Object instantiation and specification file reference during the system initialization stage.

description(GSD) comprises a stream of waveform events that occur in the current EEG scene, which is related to the detection of epileptic spikes. The time interval of the EEG scene which the GSD describes is determined by the focus interval and two parameters, Queuing-Lingering-Period and Detect-Latency-Period, as shown in Figure 4-12. In the GSD, waveform events are aligned according to the temporal order of the occurrence by using a doubly-linked list data structure. When two waveform events occur simultaneously, the event occurring in the lower channel appears early in the stream.

When there is a spike that has sufficient contextual information found in the GSD, the CRBS initiates the hypothesis-confirmation process, which is performed as follows:

- the CRBS establishes the initial hypothesis that the spike be epileptic.
- the CRBS requests the artifact SRBS to put a veto. If there is a veto, the process will be terminated with rejecting the initial hypothesis. The process without any veto will proceed.
- the CRBS requests the ssw SRBS to give its view. A positive, conclusive view lets the process be terminated with successfully confirming the initial hypothesis, while a negative one terminates the process with rejecting the hypothesis. Unless the main SRBS has a conclusive view, the process will proceed.
- the CRBS requests malpha, msigma, artifact, and meyemovement SRBSes to provide their views. After collecting the supporting and conflicting views, the CRBS resolves, if any, the conflict among them, using domain rules, and finally determines an official view regarding the initial hypothesis.

In the following sections, three main tasks performed by the CRBS and SRBSes during the hypothesis-confirmation process are described with some examples.

4.4.4 Subjective View Formation

When the CRBS needs the subjective view of a SRBS, it sends the SRBS a message to trigger the subjective view formulation process of the SRBS. The message contains the information about the focus time interval and focus channel. Upon the receipt of the message, the SRBS observes the current EEG scene from its subjective point of view and builds a spatial and a temporal scene. The spatial scene is the doubly-linked list of the focus event and its synchronous waveform events of the same type, i.e., synchronous spike

waveform events. The temporal scene is also the doubly-linked list of conflicting and supporting waveform events that contend, precede and follow the focus event. Figure 4-13 (a) and (b) show waveform event alignments in the spatial and temporal scenes, which is performed by the scene interpreter of the espike SRBS.

Using the definitions for spatial and temporal relations, the SRBS represents both the spatial and temporal scenes as a set of fact class objects. The resulting scene representation tends to be biased and subjective, because each SRBS uses different point of view to build the scenes on which the representation is based. The fact class objects have three attributes: predicate, and interpretation, and also encapsulates some methods for retrieving these attributes and for manipulating facts in a queue. The following is a subjective scene representation of the spatial and the temporal scene in Figure 4-13, made by the espike SRBS.

Fact #1
Interpretation: the focus event is a spike.
Predicate: (is this spike)

Fact #2
Interpretation: the focus event has a strong spatial support.
Predicate: (spatial-support this strong)

Fact #3
Interpretation: there is a slow wave following the focus event.
Predicate: (following this slow-wave)

Fact #4
Interpretation: the focus event has a contending alpha waveform event.
Predicate: (contending this alpha)

Fact #5
Interpretation: the focus event has a contending muscle activity.
Predicate: (contending this muscle)

Once the subjective scene representation is made, the SRBS starts its reasoning process to determine its subjective view on the current EEG scene.

4.4.5 Spatial and Temporal Context Representation

When both the main SRBS and the veto SRBSes fail to come up with a conclusive view, the main SRBS transfers the CRBS the representation of its spatial and temporal scenes, and they are stored in the Snap-Shot Fact Base and form the ground of the spatial and temporal context representation for the CRBS. Additional contextual clues are obtained from supporting and conflicting SRBSes. When a supporting SRBS gives a conclusive view, the view is represented as a fact object and added to the Snap-Shot Fact Base. If the supporting SRBS deals with precursor and the view is positive, the fact will be as follows:

Fact Support#1
Interpretation: the focus event has a supporting precursor.
Predicate: (has this supporting-precursor)

If the view is negative, the following fact will be inserted into the Snap-Shot fact base of the CRBS.

Fact Support#2
Interpretation: the focus event does not have a supporting precursor.
Predicate: (has-no this supporting-precursor)

The conclusive view of a conflicting SRBS is also represented as a fact object and added to the Snap-Shot Fact Base. The positive view of a conflicting SRBS that deals with contending events is represented as follows:

Fact Conflict#1
Interpretation: the focus event has a conflicting contender.
Predicate: (has this conflicting-contender)

If the view is negative,

Fact Conflict#2
Interpretation: the focus event does not have no conflicting contender.
Predicate: (has-no this conflicting-contender)

When either a conflicting SRBS or a supporting SRBS fails to come up with a conclusive view, one of the following fact objects is added to the Snap-Shot fact base.

Fact Support#3
Interpretation: the focus event has an unclear supporting precursor.
Predicate: (has-unclear this supporting-precursor)

or

Fact Conflict#3
Interpretation: the focus event does not have an unclear conflicting contender.
Predicate: (has-unclear this conflicting-contender)

4.4.6 Conflict Resolution and Problem Solving

As described previously, the CRBS respects the views of the main SRBS and the veto SRBSes when they have a conclusive view. If both types of the SRBSes fail to come up with a conclusive view, the CRBS makes a decision by incorporating contextual clues in the Snap-Shot Fact Base. However, the Snap-Shot Fact Base often contains both the conflicting contextual clues and the supporting contextual clues. The inference engine of the CRBS utilizes backward reasoning to resolve such conflicts and to make a final decision. The Rule Base of the CRBS consists of various rules for integrating the contextual clues and resolving the conflicts between them. As a final result, the CRBS determines whether the focus event is epileptic or not, and terminates the hypothesis-confirmation process. The focus event with a successful confirmation of the initial hypothesis is sent to the output manager and stored in a result file.

4.4.7 System Implementation

The knowledge-based contextual analysis system described above was written in C++, which is furnished with main features of object-oriented programming. Object-oriented programming centers around several major concepts: abstracts data types and classes, type hierarchies (subclass), inheritance, and polymorphism. An abstract data type is a model that encompasses a type and an associated set of operations. These operations are defined for and characterize the behavior of the underlying type. In C++, a class definition describes the behavior of the underlying abstract data types by defining the interface to all the operations that can be performed on the underlying type. The class

definition also specifies the implementation details or data structure of the type. Usually, these implementation details are accessible only within the scope of the class. We call such a type a private type. When all or parts of the data type are accessible outside the scope of the class, we call such portions of the type public. The operations that are defined for a class are called methods. These methods are analogous to procedures and functions in non–object-oriented languages. An object is a variable declared to be of a specific class. Such an object encapsulates state by containing a copy of all the fields of data (both private and public) that are defined in the class definition. Actions may be performed on such an object by invoking one or more of the methods defined in the class definition. The process of invoking a method is called sending a message to the object. Such a message typically contains parameters just as in a procedure or function call invocation in a non–object-oriented language. The invocation of a method (sending a message to an object) typically modifies the data stored in the particular object. Each class variable or object represents an instance of the class. If several objects are defined to be of the same class, they will typically contain sets of values different from each other.

To implement the knowledge-based contextual analysis system, the system is partitioned into classes, using top-down data type decomposition. In the top-down data type decomposition, a programmer first identifies and models the major data elements in the system as classes. The top-level data components (classes) are partitioned into more specialized subclasses, and this process is continued until a hierarchical structure of classes and objects is constructed. Using the top-down data type decomposition, the classes at the top of the hierarchy usually embody, through methods, characteristics that are common to all the lower sections of the hierarchy. In this study, the system is first partitioned into five classes: TimeEvent, GSD, CRBS, SRBS, and StreamManager classes. The GSD inherits the class TimeEventQueue, and both the CRBS and SRBS classes inherit an Inference class, which is further partitioned into a FactBase, a RuleBase, and a GoalStackList class. Each class encapsulates methods which characterize the behavior of the class. Both the

FactBase and the RuleBase encapsulates the following methods for manipulating their elements:

- affix — Push an element as its first element.
- append — Append an element as its last element.
- detach — Detach a specific element.
- remove — Detach a specific element and delete it.
- contain — Determine whether a specific element is contained.
- store — Store the FactBase (or RuleBase) in a file.
- load — Load all the facts (or rules) from a file.

The FactBase also has the method "lookup" to help a reasoning process.

- lookup — Ascertain whether the FactBase contains a fact matching with a certain pattern.

Since the Inference class has the FactBase and the RuleBase as its parts, it can embody all their characteristics. Besides, it encapsulates methods used for performing backward reasoning and forward reasoning. In this study, only the backward reasoning is used. The TimeEvent class, which is used to represent all kinds of EEG events, has five attributes for representing the time interval and electrode placement at which it occurs and for describing the type and name of the event. Two attributes are also reserved for making a queue. The TimeEvent encapsulates various methods for describing spatial and temporal relations.

4.5 Performance Evaluation

The developed system was tested with five sets of EEG data digitized from different subjects. This section describes the data collection and evaluation procedure. Finally, experimental results are compared with the visual screening made by two certified EEGers to evaluate the system performance.

4.5.1 Data Collection and Evaluation Procedure

In this study, six EEG records obtained from different subjects were used to test the system. The EEG collected from 16 referential scalp derivations of the 10-20 international system were first recorded on a paper for one and half hour. After reviewing the EEG on a paper, the EEG obtained from 8 referential derivations of the hemisphere that exhibits more epileptic activities than the other were digitized with 8-bit resolution and 250 Hz sampling rate, by using a Microtronics Inc. SAC computer system. The EEG was amplified by 10^4 times in order to fully use the dynamic range of the A/D converter in the SAC computer system. Since the A/D converter has 5 volt dynamic range, a quantization step corresponds to 2 uV.

The overall system for detecting epileptic spikes consists of the waveform detection system and the knowledge-based contextual analysis system, which run separately and communicate through the intermediate result file on a hard disk. The waveform detection system was configured to detect spikes, alpha waves, sigma waves, eye-movements, muscle artifacts, delta waves, and discharge waves (see Figure 4-9).

Two EEG records were used to adjust the parameters of the waveform detection system and to determine the domain rules and parameters of the knowledge-based contextual analysis system. The training process relies mainly on human visual feedback. The EEG records were processed by the waveform detection system, and its parameters were adjusted to give good machine agreements. The resulting waveform detections were applied to the knowledge-based contextual analysis system, and its parameters and rules were also tuned to give good man/machine agreements by trial and error. The finally tuned parameters of the waveform detection system are shown in APPENDIX A, while the tuned parameters and rules of the knowledge-based contextual analysis system are shown in APPENDIX B.

After tuning both the waveform detection system and the knowledge-based contextual analysis system, four EEG records were processed to evaluate the overall system

performance. The EEG records were selected to contain a variety of epileptic spikes and other abnormalities including background slowing. The patients talked, ate, and moved, so the recording included eye movement, muscle, electrode artifacts. Sleep sections were also recorded. None of the patients whose EEG was used for tuning the systems were used for the evaluation of the system performance. The waveform detection system detected spikes, alpha waves, sigma waves, eye-movements, muscle artifacts, delta waves, and discharge waves from the EEG records and passed them to the knowledge-based contextual analysis system through an intermediate result file on a hard disk. The final detections from the knowledge-based contextual analysis system were displayed on the computer screen together with the visual scorings made by one of the EEGers at a time. Some adjustments of the parameters and rules of the knowledge-based contextual analysis system were made during the test.

The agreement between the computer and an EEGer was assessed by a detection ratio and the number of false detections per minute. Agreement took place when there was an overlap between an epileptic spike marked by an EEGer and a computer detected epileptic spike. The detection ratio is defined as the number of consensus epileptic spikes (i.e., epileptic spikes marked by both EEGers) that were detected by the computer, as a percentage of the total number of consensus epileptic spikes. The false detection rate or computer over-recognitions is defined as the number of false positives made by the computer but not marked by either EEGer (i.e., isolated computer detections) per minute. Thus, the false positives refer to extra detections made by the computer and not considered important by either EEGer.

4.5.2 Man/Man Agreement

Two certified EEGers working at the same hospital were asked to mark individual epileptic spikes in the EEG paper tracings, which contain 16 channels of EEG and are one and half hours long. The EEG paper tracing contains no comment regarding the patients'

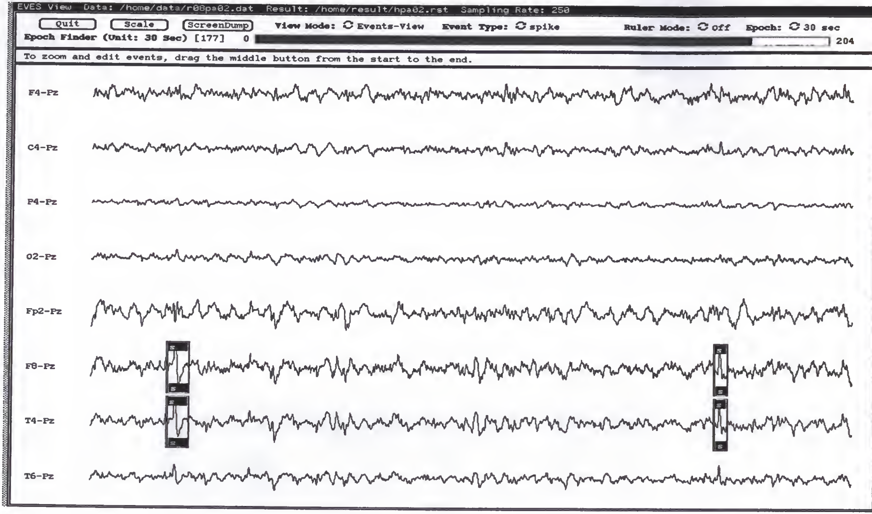


Figure 4-15. A scene of the event view of EVES.

Scorer / Subject	EEGer A	EEGer B	Both agreed
R08BU01	82	35	25
R08PA02	319	81	60
R08RO01	2	0	0
R08DO01	26	0	0

Table 4-2. Comparison of two EEGers' visual scorings (the number of individual epileptic spikes marked in one and half hour-long test period).

Scorer / Subject	EEGers agreed	System	Both agreed
R08BU01	25	4858	22
R08PA02	60	980	55
R08RO01	0	547	0
R08DO01	0	16	0

Table 4-3. Comparison of EEGer A's visual scoring and the waveform detection system's output (the number of individual epileptic spikes marked during one and half hour-long test period).

states, i.e., whether they are moving, sleeping, or chewing. The detections marked by the two EEGers in the test period are compared in Table 4-2.

The inter-EEGer disagreement was computed for one EEGer as the rate of the number of epileptic spikes marked by the reader and not by the other, to the total number of epileptic spikes marked by the reader. This inter-EEGer discrepancy must be taken into consideration when comparing the computer method to human visual assessment. In the subjects R08BU01 and R08PA02, the two EEGers did not completely agree with each other; in R08BU01, the number of epileptic spikes marked by either or both readers was 92 while they agreed only on 25 epileptic spikes, and in R08PA02, 340 epileptic spikes were marked by either or both EEGers while only 60 epileptic spikes were agreed by both EEGers. In R08RO01 and R08DO01, the EEGer B detected some epileptic spikes while the EEGer A did not detect any of them.

4.5.3 Man/Machine Agreement

The same EEG records visually screened by the two EEGers were processed by the waveform detection system, and the waveform events detected were analyzed by the knowledge-based contextual analysis system (remember that the digitized EEG records contain only 8 channels of EEG data collected from the hemisphere exhibiting more active epileptic activities, while the EEGers analyzed 16 channels of EEG data obtained from both hemispheres). The detection results were displayed on a computer screen for visual scrutiny, using EVES (see Figure 4-15), and the detection agreements were obtained by using the program "evaluate."

The detection results of the spike waveform detector are compared with those agreed by two EEGers in Table 4-3. In the subject R08BU01 the detection ratio was 88%, while 92% of epileptic spikes agreed by two EEGers were detected in the subject R08PA02. In the subject R08RO01 and R08DO01, there was no epileptic spike agreed by both

Subject \ Scorer	EEGer A	System	Both agreed
R08BU01	82	207	38
R08PA02	319	336	116
R08RO01	2	33	0
R08DO01	26	2	0

Table 4-4. Comparison of EEGer A's visual scoring and the knowledge-based contextual system's output (the number of individual epileptic spikes marked during one and half hour-long test period).

Subject \ Scorer	EEGer B	System	Both agreed
R08BU01	35	207	20
R08PA02	81	336	52
R08RO01	0	33	0
R08DO01	0	2	0

Table 4-5. Comparison of two EEGer B' visual scoring and the knowledge-based contextual analysis system's output (the number of individual epileptic spikes).

Subject \ Scorer	EEGers agreed	System	Both agreed
R08BU01	25	207	18
R08PA02	60	336	38
R08RO01	0	33	0
R08DO01	0	2	0

Table 4-6. Comparison of the detections EEGers agreed and the output of the knowledge-based contextual analysis system (the number of individual epileptic spikes).

Subject \ Scorer	EEGers agreed	System	Both agreed
R08BU01	25	3566	18
R08PA02	60	356	45
R08RO01	0	479	0
R08DO01	0	3	0

Table 4-7. Comparison of the detections EEGers agreed and the output of the knowledge-based contextual analysis system having no rule related to alpha activity (the number of individual epileptic spikes).

EEGers. Thus, the detection ratio could not be calculated. In most cases, a large number of false detections were reported.

After the knowledge-based contextual analysis system eliminates false detections, the final detection results were compared with those of EEGer A and EEGer (see Table 4-4 and 4-5), and with those agreed by both EEGers (Table 4-6). According to Table 4-6, the detection ratios of the subjects, R08BU01 and R08PA02, are 72% and 63%, respectively. In the subjects, R08RO01 and R08DO01, the detection ratio cannot be calculated because there exists any epileptic spike agreed by both EEGers. The number of false positive detections per minute is 2.08 in R08BU01 and 2.89 in R08PA02. In R08RO01 and R08DO01, in which there was no epileptic spike agreed by both EEGers, the number of false positive detections per minute is 0.33 and 0.02, respectively. As shown above, false detections generated by the spike waveform detector were dramatically suppressed, but some of the genuine detections were also rejected during the contextual analysis.

Experiments have been done to examine how much the system performance is affected by certain contextual clues. First, all the rules related to alpha activity were excluded from the knowledge-based contextual analysis system, and its detection results are shown in Table 4-7. Notice that a large number of false detections remained without using alpha-related rules in R08BU01 and R08RO01. When the rules related to muscle activity were not used, many false detections were remained in R08PA02 (see Table 4-8). The rules related to sigma activity eliminated considerable number of false detections in R08PA02 (see Table 4-9). Adding contextual clues always leads to the improvement of the system performance; a large number of false positive detections are eliminated, but some true detections were also rejected as false positive detections.

Another experiment has been done to investigate the effect of multichannel information in eliminating false positive detections. Channel 1 and Channel 2 of the subject R08BU01 were taken as reference channels. Since Channel 1 is epileptic and Channel 2 contains no epileptic spike, the effect of the multichannel information can be investigated

Subject \ Scorer	EEGers agreed	System	Both agreed
R08BU01	25	353	19
R08PA02	60	580	43
R08RO01	0	67	0
R08DO01	0	3	0

Table 4-8. Comparison of the detections EEGers agreed and the output of the knowledge-based contextual analysis system having no rule related to muscle activity (the number of individual epileptic spikes).

Subject \ Scorer	EEGers agreed	System	Both agreed
R08BU01	25	346	18
R08PA02	60	543	40
R08RO01	0	69	0
R08DO01	0	3	0

Table 4-9. Comparison of the detections EEGers agreed and the output of the knowledge-based contextual analysis system having no rule related to sigma activity (the number of individual epileptic spikes).

Channels Used \ Reference	Channel 1	Channel 2
Without using any context clues	575	32
One and Two	8	8
One through Three	4	4
One through Four	3	3
One through Five	24	1
One through Six	12	1
One through Seven	24	2
One through Eight	22	1
Both EEGer agreed	1	0

Table 4-10. The number of spike detections versus the number of channels involved in the contextual analysis.

in different situations. The experiment started with performing the contextual analysis on the two reference channels, and then continued with adding one channel at a time. Table 4-10 shows the number of spike detections versus the number of channels involved in the contextual analysis in Channel 1 and Channel 2. In the subject R08BU01, Channel 1, 5, 6, and 7 are epileptic. In Channel 1, the number of detections was decreased when Channel 3 and 4 were added, while more detections were made when epileptic channels were added. In Channel 2, less detections were made as more channels were added.

4.6 Discussion

The distributed knowledge-based system model presented here serves as a general model for contextual analysis in processing multichannel EEG data. The spatial and temporal context information of the EEG were represented by using the Interval Calculus and an object-oriented approach, and the heuristic knowledge the EEGer uses in the visual analysis was encoded into production rules. The distributed knowledge-based system model can get some advantages in the processing speed and the software design and maintenance when the heuristic knowledge consists of relatively independent multiple domain knowledge.

Based on the distributed knowledge-based system model, a contextual analysis system was developed for the elimination of false positive detections generated by the spike waveform detector. The system can be configured to process more channels and different types of contextual clues by simply modifying the parameters and rules of overall and specialized task specification files.

Advantages of automated processing of EEG data include inherent consistency of interpretation, rapid and inexpensive data reduction, and on-line EEG monitoring to initiate data storage or analysis. To be effective, the computer must accurately interpret the EEG and is more useful if they can operate in real time or faster. The low-level waveform detection system detected most of epileptic spikes marked by two EEGers, but it also made

a lot of false detections the EEGers did not mark. The knowledge-based contextual analysis system presented here eliminated a considerable amount of the false positive detections without considerably degrading detection performance. Generally, it disagreed with each EEGer more than the EEGers disagreed with each other. However, in the subjects with no epileptic spike agreed by both EEGers, most of the false positives were eliminated.

The effect of a certain contextual clue on the elimination of false positive detections is varying, depending on the type of the subject. In the subject of which the patient is sleeping, the alpha and sigma activities played an important role, while muscle activity served as a main contributor in processing awake subjects. The use of contextual information obtained from multiple channels not only eliminates more false positive detections in non-epileptic channels, but also helps to reliably detect genuine epileptic spikes.

The goal of this study is to develop an automated system that assists the EEGer, not a completely automated system that replaces the EEGer. Thus, a relatively small number of false positive detections reported here are not very serious because they can be easily identified in the subsequent visual review. Currently, the overall system operates in an off-line, batch mode, and it cannot process 8 channels of EEG data in real-time on Sun 3/50 Unix workstation. However, the processing speed can be improved by using the faster machine. An array of special purpose processors which perform the search-oriented detection of characteristic line segments in a parallel fashion could greatly enhance the processing speed.

CHAPTER 5
CONCLUSION

In this chapter, a summary of the main ideas described in this dissertation is described, and directions for further research in automated EEG analysis is suggested, especially in the problem of detecting epileptic spikes.

5.1 Summary of Main Ideas

This dissertation describes the development of an automated EEG analysis system that assist the EEGer in prescreening epileptic patients. The design concentrates on the problem of detecting epileptic inter-ictal spikes. Two different efforts are explored to relieve the obstacle that hampers the clinical application of currently existing automated EEG analysis systems.

Firstly, computer-based tools for displaying the results of automated analysis systems together with the EEG were developed. Through the use of these window-based, graphic menu-driven visualization tools, the user can search, mark, and review portions of EEG data selected by the user or processed by the automated analysis system. These tools are also used for the design and evaluation of filters and waveform detection algorithms.

Secondly, a new waveform detection algorithm was developed to improve the detection performance of automated EEG analysis. The new morphological waveform detection algorithm first detects characteristic line segments through a guided search, and then performs a structural analysis on a sequence of the characteristic line segments detected. The line segment sequence that meets given structural features is classified as a waveform of interest. Known characteristics of EEG waveform morphology are used to limit the search space. The use of a search technique in detecting characteristic line

segments needs more elaborate structural analysis since adjacent line segments obtained from the search are not always connected together. However, the search process brings in some useful properties. Even when a fast activity is superimposed over a slow activity, the algorithm can usually detect the fast activity without a priori signal conditioning, such as band-pass filtering, since it is based on the search for line segments satisfying predefined requirements. Thus, the filter design step can also be skipped when developing waveform detectors that use the waveform detection algorithm presented here. The unwanted waveform distortions due to the filtering can also be avoided. Moreover, since each line segment search process operate independently, the processing speed can be accelerated by using multiple processors operating in a parallel fashion. In other words, the algorithm is appropriate for parallel processing. The waveform detection system implemented here is based on the search-oriented waveform detection algorithm and the unified waveform specification format. The system is configurable in the sense that its operation is entirely relies on the specifications provided by the user, without any modification of the source code. Thus, different waveform detectors can be obtained by simply switching the input specification files. In the experiments, the system gave results comparable with those obtained from one of the most reliable automated methods in the detection of sleep EEG waveforms.

Finally, a knowledge-based contextual analysis system was developed to eliminate false detections generated by the waveform detection system. The knowledge-based system analyzes various contextual clues by using the encoded heuristic knowledge of the EEGer. A distributed knowledge-based system model was used to simulate the confirmatory and counteractive interactions of contextual clues in the visual EEG analysis, which is called a hypothesis-confirmation process. The distributed knowledge-based system model consists of many specialized rule-based systems that cooperate and compete together and a coordinating rule-based system that supervises the specialized rule-based systems. The coordinating rule-based system establishes an initial hypothesis, collects supporting and

conflicting contextual clues from specialized rule-based systems, and finally determines whether to confirm or to reject the initial hypothesis. The distributed knowledge-based system model improves the processing speed and facilitates the software design and maintenance in solving the problem that is readily divided into relatively independent, multiple subproblems. The knowledge-based contextual analysis system is extensible in the sense that it can be configured by the external configuration file without modifying the source code.

The automated system for detecting epileptic spikes consists of the low-level waveform detection system for the detection of spike, alpha, sigma, slow waves, eye-movements and muscle activity, and the high-level knowledge-based contextual analysis system for the elimination of false positive detections generated by the spike waveform detector. The low-level waveform detection system detects waveform events relevant to epileptic spikes from the EEG and constructs a stream of waveform events. A portion of EEG data which contain sufficient information for contextual analysis is called an EEG scene. The high-level knowledge-based contextual analysis system accepts and analyzes the EEG scene and eliminates false positive detections. Experimental results show that the automated epileptic spike detection system performs well in detecting epileptic spikes. However, it still suffers from false positive detections, mainly due to epileptic sharp activities. In two subjects with epilepsy, the system showed 72% and 63% detection performance, respectively. The false positive detection rates were 2.08 and 2.89 false positives per minute. In two subjects with no epileptic spike agreed by two EEGers, the system gave 0.02 and 0.33 false positive detections per minute.

Experimental results showed that the sensitivity of the system performance to a certain contextual clue deeply relies on the type of the subject. In sleep subjects, the alpha and sigma activities played an important role in the elimination of false positive detections, while muscle activities served as a main contributor in awake subjects. The experimental results also showed that the use of the contextual information obtained from multiple

channels not only helps to eliminate more false positive detections in non-epileptic channels, but also helps to reliably detect genuine epileptic spikes.

5.2 Further Work

The computer-based visualization tools developed here were proved to be useful in the design and evaluation of filters and signal detection algorithms. However, the tools need to be improved in speed and screen resolution for more comfortable clinical application. The rapid progress of display and storage technology will make it possible to build a completely computerized EEG workstation in a near future.

The EEG scene description can be improved by extending the low-level waveform detection system to detect additional contextual clues. The state of consciousness of the subject, the trends of amplitude and frequency of background EEG, the regularity of the various EEG waveforms and their rhythmic nature, and variations across several electrodes on the same or the other hemisphere can serve as important contextual clues in eliminating false positive detections, if they can be represented in a form that can be manipulated by the computer. The refinement of parameter sets used in the waveform detection is also another way of enhancing the EEG scene description. The use of more heuristic knowledge could lead to the elimination of more false positive detections without missing many genuine epileptic spikes.

APPENDIX A
WAVEFORM MODEL SPECIFICATION

Several different types of EEG waveforms were involved in the detection of epileptic spikes from normal EEG activity. The existence of delta activity in the temporal proximity of a spike waveform served as an important contextual clue in determining whether the spike is epileptic or not. Other activities, such as alpha, sigma, muscle, and eye-movement activities, were taken into account to eliminate false positive detections, mainly due to insufficient discriminatory power of morphological parameters of the spike waveform detector.

Waveform detectors are implemented as objects of a same class, which encapsulate the search-oriented waveform detection algorithm as the method. The operation of each object entirely depends on the waveform model specification provided by the user, specifying the characteristic line segments and their structural features. A line segment model specification file contains a set of parameter values needed for detecting a specific characteristic line segment, while a waveform model specification file contains encoded heuristic domain knowledge and a set of parameter values specifying the structural features of a sequence of characteristic line segments that can be classified as a specific waveform. The waveform model specification files are named after the corresponding waveform name and have the extension "wave," while the line segment model specification files have the extension "seg."

In this part, waveform models of seven EEG waveforms related to the detection of epileptic spikes are shown.

1. Spike Waveform Model Specification

The spike waveform consists of two different characteristic line segments: p1sp and n2sp. In the specification files, the symbol "//" and a pair of the symbols "/*" and "*/" are used as comment symbols.

```
////////////////////////////////////////////////////////////////////////////////////////
//              File Name: spike.wave                                                 //
//              Date: October 31, 1990.                                               //
//              Comment:   this is the waveform model                                 //
//                         specification file for epileptic spikes.                   //
////////////////////////////////////////////////////////////////////////////////////////
  Waveform-Type           :       Single-Phasic
  Waveform-Name           :       spike
  Interval-Balance-Rate   :       15
  Amplitude-Change-Rate   :       30
  Number-Of-Wave-Models   :       1

////////////////////////////////////////////////////////////////////////////////////////
//              Waveform model for Uprising Spikes                                    //
////////////////////////////////////////////////////////////////////////////////////////
//              p1sp --------- the first positive line segment.                       //
//              n2sp --------- the second negative line segment.                      //
////////////////////////////////////////////////////////////////////////////////////////
  PatternDefinition
    Waveform-Model-Name               :   uprising-spike
    Minimum-Total-Duration            :   30      // (unit:msec)
    Maximum-Total-Duration            :   110     // (unit:msec)
    Average-Duty-Ratio                :   70      // (unit:%)
    Sequential-Segment-Pattern        :   <p1sp><n2sp>
    Minimum-Single-Wave-Period        :   <30>  // (unit:msec)
    Maximum-Single-Wave-Period        :   <110> // (unit:msec)
    Minimum-Single-Wave-Duty-Ratio    :   <70>  // (unit:%)
  EndDefinition
////////////////////////////////////////////////////////////////////////////////////////
```

```
//////////////////////////////////////////////////////////////////////////////////////////////
//              File Name: p1sp.seg                                                         //
//              Date: October 31, 1990.                                                     //
//              Comment:   this is the line segment model                                   //
//                         for epileptic spike detection.                                   //
//////////////////////////////////////////////////////////////////////////////////////////////
//              Segment Name       :      p1sp                                              //
//              Time unit          :      millisec                                          //
//              Amplitude unit     :      sVolt (0.1 uVolt)                                 //
//              Slope unit         :      0.1 sVolt / millisecond.                          //
//              In each Contraction-Rate, 50% is full range.                                //
//              Signal-To-Noise-Ratio is a signal / noise.                                  //
//////////////////////////////////////////////////////////////////////////////////////////////
  Upper-Contraction-Rate   :    6
  Lower-Contraction-Rate   :    6
  Minimum-Period           :    12
  Maximum-Period           :    40
  Minimum-Amplitude        :    650
  Maximum-Amplitude        :    5000
  Minimum-Slope            :    350
  Maximum-Slope            :    2500
  Slope-Direction          :    Positive
  Selection-Policy         :    Max-Amplitude
//////////////////////////////////////////////////////////////////////////////////////////////
```

```
//////////////////////////////////////////////////////////////////////////////////////////////
//              File Name: n2sp.seg                                                         //
//              Date: October 31, 1990.                                                     //
//              Comment:   this is the line segment model                                   //
//                         for epileptic spike detection.                                   //
//////////////////////////////////////////////////////////////////////////////////////////////
//              Segment Name       :      n2sp                                              //
//              Time unit          :      millisec                                          //
//              Amplitude unit     :      sVolt (0.1 uVolt)                                 //
//              Slope unit         :      0.1 sVolt / millisecond.                          //
//              In each Contraction-Rate, 50% is full range.                                //
//              Signal-To-Noise-Ratio is a signal / noise.                                  //
//////////////////////////////////////////////////////////////////////////////////////////////
  Upper-Contraction-Rate   :    12
  Lower-Contraction-Rate   :    10
  Minimum-Period           :    20
  Maximum-Period           :    70
  Minimum-Amplitude        :    2000
  Maximum-Amplitude        :    6000
  Minimum-Slope            :    200
  Maximum-Slope            :    2000
  Slope-Direction          :    Negative
  Selection-Policy         :    Max-Amplitude
//////////////////////////////////////////////////////////////////////////////////////////////
```

2. Delta Waveform Model Specification

The delta waveform model contains two patterns, upward-delta and downward-delta, both consist of two characteristic line segments, pcdl and ncdl.

```
//////////////////////////////////////////////////////////////////////////////////////////////
//            File Name: delta.wave                                        //
//            Date: October 31, 1990.                                      //
//            Comment:   this is the waveform model                        //
//                       specification file for delta (slow) waves.        //
//////////////////////////////////////////////////////////////////////////////////////////////
  Waveform-Type              :      Single-Phasic
  Waveform-Name              :      delta
  Interval-Balance-Rate      :      65
  Amplitude-Change-Rate      :      65
  Number-Of-Wave-Models      :      2

//////////////////////////////////////////////////////////////////////////////////////////////
//            Waveform Model for Delta Waves                               //
//////////////////////////////////////////////////////////////////////////////////////////////
//            pcdl --------- the positive line segment.                    //
//            ncdl --------- the negative line segment.                    //
//////////////////////////////////////////////////////////////////////////////////////////////
  PatternDefinition
    Waveform-Model-Name                :   upward-delta
    Minimum-Total-Duration             :   240        // (unit:msec)
    Maximum-Total-Duration             :   1000       // (unit:msec)
    Average-Duty-Ratio                 :   80         // (unit:%)
    Sequential-Segment-Pattern         :   <pcdl><ncdl>
    Minimum-Single-Wave-Period         :   <240>      // (unit:msec)
    Maximum-Single-Wave-Period         :   <1000>     // (unit:msec)
    Minimum-Single-Wave-Duty-Ratio     :   <80>       // (unit:%)
  EndDefinition

  PatternDefinition
    Waveform-Model-Name                :   downward-delta
    Minimum-Total-Duration             :   240        // (unit:msec)
    Maximum-Total-Duration             :   1000       // (unit:msec)
    Average-Duty-Ratio                 :   80         // (unit:%)
    Sequential-Segment-Pattern         :   <ncdl><pcdl>
    Minimum-Single-Wave-Period         :   <240>      // (unit:msec)
    Maximum-Single-Wave-Period         :   <1000>     // (unit:msec)
    Minimum-Single-Wave-Duty-Ratio     :   <80>       // (unit:%)
  EndDefinition
//////////////////////////////////////////////////////////////////////////////////////////////
```

```
///////////////////////////////////////////////////////////////////////////////////////////
//              File Name: pcdl.seg                                                    //
//              Date: October 31, 1990.                                                //
//              Comment:    this is the line segment model                             //
//                          for delta (slow) wave detection.                           //
///////////////////////////////////////////////////////////////////////////////////////////
//              Segment Name         :        pcdl                                     //
//              Time unit            :        millisec                                 //
//              Amplitude unit       :        sVolt (0.1 uVolt)                        //
//              Slope unit           :        0.1 sVolt / millisecond.                 //
//              In each Contraction-Rate, 50% is full range.                           //
//              Signal-To-Noise-Ratio is a signal / noise.                             //
///////////////////////////////////////////////////////////////////////////////////////////
  Upper-Contraction-Rate    :      15
  Lower-Contraction-Rate    :      15
  Minimum-Period            :      40
  Maximum-Period            :      200
  Minimum-Amplitude         :      600
  Maximum-Amplitude         :      3000
  Minimum-Slope             :      40
  Maximum-Slope             :      700
  Slope-Direction           :      Positive
  Selection-Policy          :      Max-Time
///////////////////////////////////////////////////////////////////////////////////////////


///////////////////////////////////////////////////////////////////////////////////////////
//              File Name: ncdl.seg                                                    //
//              Date: October 31, 1990.                                                //
//              Comment:    this is the line segment model                             //
//                          for delta (slow) wave detection.                           //
///////////////////////////////////////////////////////////////////////////////////////////
//              Segment Name         :        ncsp                                     //
//              Time unit            :        millisec                                 //
//              Amplitude unit       :        sVolt (0.1 uVolt)                        //
//              Slope unit           :        0.1 sVolt / millisecond.                 //
//              In each Contraction-Rate, 50% is full range.                           //
//              Signal-To-Noise-Ratio is a signal / noise.                             //
///////////////////////////////////////////////////////////////////////////////////////////
  Upper-Contraction-Rate    :      20
  Lower-Contraction-Rate    :      20
  Minimum-Period            :      70
  Maximum-Period            :      200
  Minimum-Amplitude         :      600
  Maximum-Amplitude         :      3000
  Minimum-Slope             :      40
  Maximum-Slope             :      450
  Slope-Direction           :      Negative
  Selection-Policy          :      Max-Amplitude
///////////////////////////////////////////////////////////////////////////////////////////
```

3. Discharge Waveform Model Specification

The discharge waveform is the waveform that follows the spike. It contains only one pattern which consists of two characteristic line segments, n1di and p2di.

```
////////////////////////////////////////////////////////////////////////////////
//              File Name: discharge.wave                                     //
//              Date: October 31, 1990.                                       //
//              Comment:   this is the waveform model                         //
//                         specification file for discharge waves.            //
////////////////////////////////////////////////////////////////////////////////
  Waveform-Type            :      Single-Phasic
  Waveform-Name            :      discharge
  Interval-Balance-Rate    :      30
  Amplitude-Change-Rate    :      40
  Number-Of-Wave-Models    :      1

////////////////////////////////////////////////////////////////////////////////
//              Waveform Model for Discharge Waves                            //
////////////////////////////////////////////////////////////////////////////////
//              n1di --------- the positive line segment.                     //
//              p2di --------- the negative line segment.                     //
////////////////////////////////////////////////////////////////////////////////
  PatternDefinition
   Waveform-Model-Name                :  upward-discharge
   Minimum-Total-Duration             :  80          // (unit:msec)
   Maximum-Total-Duration             :  300         // (unit:msec)
   Average-Duty-Ratio                 :  70          // (unit:%)
   Sequential-Segment-Pattern         :  <n1di><p2di>
   Minimum-Single-Wave-Period         :  <80>        // (unit:msec)
   Maximum-Single-Wave-Period         :  <300>       // (unit:msec)
   Minimum-Single-Wave-Duty-Ratio     :  <70>        // (unit:%)
  EndDefinition
////////////////////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////////////////////////////////
//              File Name: n1di.seg                                                       //
//              Date: October 31, 1990.                                                   //
//              Comment:   this is the line segment model                                 //
//                         for discharge detection.                                       //
////////////////////////////////////////////////////////////////////////////////////////////
//              Segment Name        :      n1di                                           //
//              Time unit           :      millisec                                       //
//              Amplitude unit      :      sVolt (0.1 uVolt)                              //
//              Slope unit          :      0.1 sVolt / millisecond.                       //
//              In each Contraction-Rate, 50% is full range.                              //
//              Signal-To-Noise-Ratio is a signal / noise.                                //
////////////////////////////////////////////////////////////////////////////////////////////
  Upper-Contraction-Rate    :    15
  Lower-Contraction-Rate    :    15
  Minimum-Period            :    20
  Maximum-Period            :    70
  Minimum-Amplitude         :    2000
  Maximum-Amplitude         :    6000
  Minimum-Slope             :    200
  Maximum-Slope             :    2000
  Slope-Direction           :    Negative
  Selection-Policy          :    Max-Amplitude
////////////////////////////////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////////////////////////////////
//              File Name: p2di.seg                                                       //
//              Date: October 31, 1990.                                                   //
//              Comment:   this is the line segment model                                 //
//                         for delta (slow) wave detection.                               //
////////////////////////////////////////////////////////////////////////////////////////////
//              Segment Name        :      p2di                                           //
//              Time unit           :      millisec                                       //
//              Amplitude unit      :      sVolt (0.1 uVolt)                              //
//              Slope unit          :      0.1 sVolt / millisecond.                       //
//              In each Contraction-Rate, 50% is full range.                              //
//              Signal-To-Noise-Ratio is a signal / noise.                                //
////////////////////////////////////////////////////////////////////////////////////////////
  Upper-Contraction-Rate    :    15
  Lower-Contraction-Rate    :    15
  Minimum-Period            :    50
  Maximum-Period            :    200
  Minimum-Amplitude         :    600
  Maximum-Amplitude         :    3000
  Minimum-Slope             :    40
  Maximum-Slope             :    700
  Slope-Direction           :    Positive
  Selection-Policy          :    Max-Amplitude
////////////////////////////////////////////////////////////////////////////////////////////
```

4. Alpha Waveform Model Specification

The alpha waveform model contains two patterns, up-start-alpha and down-start-alpha, both consist of two characteristic line segments, pcal and ncal.

```
////////////////////////////////////////////////////////////////////////////////////////////////////
//              File Name: alpha.wave                                    //
//              Date: October 31, 1990.                                  //
//              Comment:   this is the waveform model                    //
//                         specification file for alpha spindles.        //
////////////////////////////////////////////////////////////////////////////////////////////////////
 Waveform-Type            :      Spindles
 Waveform-Name            :      alpha
 Interval-Balance-Rate    :      35
 Amplitude-Change-Rate    :      20
 Number-Of-Wave-Models    :      2
////////////////////////////////////////////////////////////////////////////////////////////////////
//              Waveform Model for Alpha Spindles                        //
////////////////////////////////////////////////////////////////////////////////////////////////////
//              pcal --------- the positive line segment.                //
//              ncal --------- the negative line segment.                //
////////////////////////////////////////////////////////////////////////////////////////////////////
 PatternDefinition
  Waveform-Model-Name                  :   down-start-alpha
  Minimum-Total-Duration               :   320  // (unit:msec)
  Maximum-Total-Duration               :   620  // (unit:msec)
  Average-Duty-Ratio                   :   60   // (unit:%)
  Sequential-Segment-Pattern           :   <ncal><pcal><ncal><pcal>
                                           <ncal><pcal><ncal><pcal>
  Minimum-Single-Wave-Period           :   <72><72><72><72><72>
                                           <72><72>    // (unit:msec)
  Maximum-Single-Wave-Period           :   <172><172><172><172>
                                           <172><172><172>
  Minimum-Single-Wave-Duty-Ratio       :   <50><50><50><50><50>
                                           <50><50>    // (unit:%)
 EndDefinition
 PatternDefinition
  Waveform-Model-Name                  :   up-start-alpha
  Minimum-Total-Duration               :   320  // (unit:msec)
  Maximum-Total-Duration               :   620  // (unit:msec)
  Average-Duty-Ratio                   :   60   // (unit:%)
  Sequential-Segment-Pattern           :   <pcal><ncal><pcal><ncal>
                                           <pcal><ncal><pcal><ncal>
  Minimum-Single-Wave-Period           :   <72><72><72><72><72>
                                           <72><72>    // (unit:msec)
  Maximum-Single-Wave-Period           :   <172><172><172><172>
                                           <172><172><172>
  Minimum-Single-Wave-Duty-Ratio       :   <50><50><50><50><50>
                                           <50><50>    // (unit:%)
 EndDefinition
////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////////////////////////////
//              File Name: pcal.seg                                                   //
//              Date: October 31, 1990.                                               //
//              Comment:   this is the line segment model                             //
//                         for alpha spindle detection.                               //
////////////////////////////////////////////////////////////////////////////////////////
//              Segment Name       :      pcal                                        //
//              Time unit          :      millisec                                    //
//              Amplitude unit     :      sVolt (0.1 uVolt)                           //
//              Slope unit         :      0.1 sVolt / millisecond.                    //
//              In each Contraction-Rate, 50% is full range.                          //
//              Signal-To-Noise-Ratio is a signal / noise.                            //
////////////////////////////////////////////////////////////////////////////////////////
  Upper-Contraction-Rate   :   10
  Lower-Contraction-Rate   :   20
  Minimum-Period           :   30
  Maximum-Period           :   80
  Minimum-Amplitude        :   300
  Maximum-Amplitude        :   400
  Minimum-Slope            :   50
  Maximum-Slope            :   800
  Slope-Direction          :   Positive
  Selection-Policy         :   Max-Amplitude
////////////////////////////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////////////////////////////
//              File Name: ncal.seg                                                   //
//              Date: October 31, 1990.                                               //
//              Comment:   this is the line segment model                             //
//                         for alpha spindle detection.                               //
////////////////////////////////////////////////////////////////////////////////////////
//              Segment Name       :      ncal                                        //
//              Time unit          :      millisec                                    //
//              Amplitude unit     :      sVolt (0.1 uVolt)                           //
//              Slope unit         :      0.1 sVolt / millisecond.                    //
//              In each Contraction-Rate, 50% is full range.                          //
//              Signal-To-Noise-Ratio is a signal / noise.                            //
////////////////////////////////////////////////////////////////////////////////////////
  Upper-Contraction-Rate   :   20
  Lower-Contraction-Rate   :   10
  Minimum-Period           :   30
  Maximum-Period           :   80
  Minimum-Amplitude        :   300
  Maximum-Amplitude        :   4000
  Minimum-Slope            :   50
  Maximum-Slope            :   800
  Slope-Direction          :   Negative
  Selection-Policy         :   Max-Amplitude
////////////////////////////////////////////////////////////////////////////////////////
```

5. Sigma Waveform Model Specification

The sigma waveform model contains two patterns, up-start-sigma and down-start-sigma, both consist of two characteristic line segments, pcsg and ncsg.

```
////////////////////////////////////////////////////////////////////////////////
//            File Name: sigma.wave                                          //
//            Date: Octorber 31, 1990.                                       //
//            Comment:  this is the waveform model                           //
//                      specification file for sigma spindles.               //
////////////////////////////////////////////////////////////////////////////////
 Waveform-Type           :      Spindles
 Waveform-Name           :      sigma
 Interval-Balance-Rate   :      40
 Amplitude-Change-Rate   :      20
 Number-Of-Wave-Models   :      2
////////////////////////////////////////////////////////////////////////////////
//            Waveform Model for Sigma Spindles                              //
////////////////////////////////////////////////////////////////////////////////
//            pcsg --------- the positive line segment.                      //
//            ncsg --------- the negative line segment.                      //
////////////////////////////////////////////////////////////////////////////////
 PatternDefinition
  Waveform-Model-Name             :   down-start-sigma
  Minimum-Total-Duration          :   188  // (unit:msec)
  Maximum-Total-Duration          :   292  // (unit:msec)
  Average-Duty-Ratio              :   70   // (unit:%)
  Sequential-Segment-Pattern      :   <ncsg><pcsg><ncsg><pcsg>
                                      <ncsg><pcsg><ncsg>
  Minimum-Single-Wave-Period      :   <48><48><48><48><48>
                                      <48>          // (unit:msec)
  Maximum-Single-Wave-Period      :   <96><96><96><96><96>
                                      <96>          // (unit:msec)
  Minimum-Single-Wave-Duty-Ratio  :   <55><55><55><55><55>
                                      <55>          // (unit:%)
 EndDefinition
 PatternDefinition
  Waveform-Model-Name             :   up-start-sigma
  Minimum-Total-Duration          :   188  // (unit:msec)
  Maximum-Total-Duration          :   292  // (unit:msec)
  Average-Duty-Ratio              :   70   // (unit:%)
  Sequential-Segment-Pattern      :   <pcsg><ncsg><pcsg><ncsg>
                                      <pcsg><ncsg><ncsg>
  Minimum-Single-Wave-Period      :   <48><48><48><48><48>
                                      <48>          // (unit:msec)
  Maximum-Single-Wave-Period      :   <96><96><96><96><96>
                                      <96>          // (unit:msec)
  Minimum-Single-Wave-Duty-Ratio  :   <55><55><55><55><55>
                                      <55>          // (unit:%)
 EndDefinition
////////////////////////////////////////////////////////////////////////////////
```

```
//////////////////////////////////////////////////////////////////////////////////////
//              File Name: pcsg.seg                                                //
//              Date: October 31, 1990.                                            //
//              Comment:   this is the line segment model                          //
//                         for sigma spindle detection.                            //
//////////////////////////////////////////////////////////////////////////////////////
//              Segment Name        :       pcsg                                   //
//              Time unit           :       millisec                               //
//              Amplitude unit      :       sVolt (0.1 uVolt)                      //
//              Slope unit          :       0.1 sVolt / millisecond.               //
//              In each Contraction-Rate, 50% is full range.                       //
//              Signal-To-Noise-Ratio is a signal / noise.                         //
//////////////////////////////////////////////////////////////////////////////////////
  Upper-Contraction-Rate    :     8
  Lower-Contraction-Rate    :     8
  Minimum-Period            :     16
  Maximum-Period            :     40
  Minimum-Amplitude         :     150
  Maximum-Amplitude         :     1000
  Minimum-Slope             :     40
  Maximum-Slope             :     550
  Slope-Direction           :     Positive
  Selection-Policy          :     Max-Amplitude
//////////////////////////////////////////////////////////////////////////////////////
```

```
//////////////////////////////////////////////////////////////////////////////////////
//              File Name: ncsg.seg                                                //
//              Date: October 31, 1990.                                            //
//              Comment:   this is the line segment model                          //
//                         for sigma spindle detection.                            //
//////////////////////////////////////////////////////////////////////////////////////
//              Segment Name        :       ncsg                                   //
//              Time unit           :       millisec                               //
//              Amplitude unit      :       sVolt (0.1 uVolt)                      //
//              Slope unit          :       0.1 sVolt / millisecond.               //
//              In each Contraction-Rate, 50% is full range.                       //
//              Signal-To-Noise-Ratio is a signal / noise.                         //
//////////////////////////////////////////////////////////////////////////////////////
  Upper-Contraction-Rate    :     8
  Lower-Contraction-Rate    :     8
  Minimum-Period            :     16
  Maximum-Period            :     40
  Minimum-Amplitude         :     150
  Maximum-Amplitude         :     1000
  Minimum-Slope             :     40
  Maximum-Slope             :     550
  Slope-Direction           :     Negative
  Selection-Policy          :     Max-Amplitude
//////////////////////////////////////////////////////////////////////////////////////
```

6. Eyemovement Waveform Model Specification

The eyemovement waveform model contains two patterns, up-start-eyemove and down-start-eyemove.

```
//////////////////////////////////////////////////////////////////////////////////////////////
//              File Name: eyemove.wave                                                    //
//              Date: October 31, 1990.                                                    //
//              Comment:   this is the waveform model                                      //
//                         specification file for eye-movements.                           //
//////////////////////////////////////////////////////////////////////////////////////////////
  Waveform-Type              :     Poly-Phasic
  Waveform-Name              :     eyemove
  Interval-Balance-Rate      :     10
  Amplitude-Change-Rate      :     10
  Number-Of-Wave-Models      :     2
//////////////////////////////////////////////////////////////////////////////////////////////
//              Waveform Model for Eye-Movements                                           //
//////////////////////////////////////////////////////////////////////////////////////////////
//              p1em --------- the first positive line segment.                            //
//              n2em --------- the second negative line segment.                           //
//              p3em --------- the third positive line segment.                            //
//              n1em --------- the first negative line segment.                            //
//              p2em --------- the second positive line segment.                           //
//              n3em --------- the third negative line segment.                            //
//////////////////////////////////////////////////////////////////////////////////////////////
  PatternDefinition
    Waveform-Model-Name                :   up-start-eyemove
    Minimum-Total-Duration             :   350  // (unit:msec)
    Maximum-Total-Duration             :   1000 // (unit:msec)
    Average-Duty-Ratio                 :   65   // (unit:%)
    Sequential-Segment-Pattern         :   <p1em><n2em><p3em>
    Minimum-Single-Wave-Period         :   <150><180>  // (unit:msec)
    Maximum-Single-Wave-Period         :   <500><600>  // (unit:msec)
    Minimum-Single-Wave-Duty-Ratio     :   <70><50>    // (unit:%)
  EndDefinition
  PatternDefinition
    Waveform-Model-Name                :   down-start-eyemove
    Minimum-Total-Duration             :   350  // (unit:msec)
    Maximum-Total-Duration             :   1000 // (unit:msec)
    Average-Duty-Ratio                 :   65   // (unit:%)
    Sequential-Segment-Pattern         :   <n1em><p2em><n3em>
    Minimum-Single-Wave-Period         :   <150><180>  // (unit:msec)
    Maximum-Single-Wave-Period         :   <500><600>  // (unit:msec)
    Minimum-Single-Wave-Duty-Ratio     :   <70><50>    // (unit:%)
  EndDefinition
//////////////////////////////////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////////////////////
//            File Name: p1em.seg                                             //
//            Date: October 31, 1990.                                         //
//            Comment:   this is the line segment model                       //
//                       for eye-movement detection.                          //
////////////////////////////////////////////////////////////////////////////////
//            Segment Name        :      p1em                                 //
//            Time unit           :      millisec                             //
//            Amplitude unit      :      sVolt (0.1 uVolt)                    //
//            Slope unit          :      0.1 sVolt / millisecond.             //
//            In each Contraction-Rate, 50% is full range.                    //
//            Signal-To-Noise-Ratio is a signal / noise.                      //
////////////////////////////////////////////////////////////////////////////////
  Upper-Contraction-Rate   :     12
  Lower-Contraction-Rate   :     13
  Minimum-Period           :     20
  Maximum-Period           :     200
  Minimum-Amplitude        :     600
  Maximum-Amplitude        :     7000
  Minimum-Slope            :     50
  Maximum-Slope            :     1000
  Slope-Direction          :     Positive
  Selection-Policy         :     Max-Amplitude
////////////////////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////////////////////
//            File Name: p2em.seg                                             //
//            Date: October 31, 1990.                                         //
//            Comment:   this is the line segment model                       //
//                       for eye-movement detection.                          //
////////////////////////////////////////////////////////////////////////////////
//            Segment Name        :      p2em                                 //
//            Time unit           :      millisec                             //
//            Amplitude unit      :      sVolt (0.1 uVolt)                    //
//            Slope unit          :      0.1 sVolt / millisecond.             //
//            In each Contraction-Rate, 50% is full range.                    //
//            Signal-To-Noise-Ratio is a signal / noise.                      //
////////////////////////////////////////////////////////////////////////////////
  Upper-Contraction-Rate   :     15
  Lower-Contraction-Rate   :     15
  Minimum-Period           :     60
  Maximum-Period           :     300
  Minimum-Amplitude        :     1500
  Maximum-Amplitude        :     7000
  Minimum-Slope            :     100
  Maximum-Slope            :     1000
  Slope-Direction          :     Positive
  Selection-Policy         :     Max-Amplitude
////////////////////////////////////////////////////////////////////////////////
```

```
/////////////////////////////////////////////////////////////////////////////////
//              File Name: p3em.seg                                            //
//              Date: October 31, 1990.                                        //
//              Comment:   this is the line segment model                      //
//                         for eye-movement detection.                         //
/////////////////////////////////////////////////////////////////////////////////
//              Segment Name        :       p3em                               //
//              Time unit           :       millisec                           //
//              Amplitude unit      :       sVolt (0.1 uVolt)                  //
//              Slope unit          :       0.1 sVolt / millisecond.           //
//              In each Contraction-Rate, 50% is full range.                   //
//              Signal-To-Noise-Ratio is a signal / noise.                     //
/////////////////////////////////////////////////////////////////////////////////
  Upper-Contraction-Rate   :    20
  Lower-Contraction-Rate   :    20
  Minimum-Period           :    40
  Maximum-Period           :    300
  Minimum-Amplitude        :    500
  Maximum-Amplitude        :    7000
  Minimum-Slope            :    20
  Maximum-Slope            :    1000
  Slope-Direction          :    Positive
  Selection-Policy         :    Max-Amplitude
/////////////////////////////////////////////////////////////////////////////////
```

```
/////////////////////////////////////////////////////////////////////////////////
//              File Name: n1em.seg                                            //
//              Date: October 31, 1990.                                        //
//              Comment:   this is the line segment model                      //
//                         for eye-movement detection.                         //
/////////////////////////////////////////////////////////////////////////////////
//              Segment Name        :       n1em                               //
//              Time unit           :       millisec                           //
//              Amplitude unit      :       sVolt (0.1 uVolt)                  //
//              Slope unit          :       0.1 sVolt / millisecond.           //
//              In each Contraction-Rate, 50% is full range.                   //
//              Signal-To-Noise-Ratio is a signal / noise.                     //
/////////////////////////////////////////////////////////////////////////////////
  Upper-Contraction-Rate   :    12
  Lower-Contraction-Rate   :    13
  Minimum-Period           :    20
  Maximum-Period           :    200
  Minimum-Amplitude        :    600
  Maximum-Amplitude        :    7000
  Minimum-Slope            :    50
  Maximum-Slope            :    1000
  Slope-Direction          :    Negative
  Selection-Policy         :    Max-Amplitude
/////////////////////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////////////////////////////////
//              File Name: n2em.seg                                                      //
//              Date: October 31, 1990.                                                  //
//              Comment:   this is the line segment model                                //
//                         for eye-movement detection.                                   //
////////////////////////////////////////////////////////////////////////////////////////////
//              Segment Name          :        n2em                                      //
//              Time unit             :        millisec                                  //
//              Amplitude unit        :        sVolt (0.1 uVolt)                         //
//              Slope unit            :        0.1 sVolt / millisecond.                  //
//              In each Contraction-Rate, 50% is full range.                             //
//              Signal-To-Noise-Ratio is a signal / noise.                               //
////////////////////////////////////////////////////////////////////////////////////////////
  Upper-Contraction-Rate   :      15
  Lower-Contraction-Rate   :      15
  Minimum-Period           :      60
  Maximum-Period           :      300
  Minimum-Amplitude        :      1500
  Maximum-Amplitude        :      7000
  Minimum-Slope            :      100
  Maximum-Slope            :      1000
  Slope-Direction          :      Negative
  Selection-Policy         :      Max-Amplitude
////////////////////////////////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////////////////////////////////
//              File Name: n3em.seg                                                      //
//              Date: October 31, 1990.                                                  //
//              Comment:   this is the line segment model                                //
//                         for eye-movement detection.                                   //
////////////////////////////////////////////////////////////////////////////////////////////
//              Segment Name          :        n3em                                      //
//              Time unit             :        millisec                                  //
//              Amplitude unit        :        sVolt (0.1 uVolt)                         //
//              Slope unit            :        0.1 sVolt / millisecond.                  //
//              In each Contraction-Rate, 50% is full range.                             //
//              Signal-To-Noise-Ratio is a signal / noise.                               //
////////////////////////////////////////////////////////////////////////////////////////////
  Upper-Contraction-Rate   :      20
  Lower-Contraction-Rate   :      20
  Minimum-Period           :      40
  Maximum-Period           :      300
  Minimum-Amplitude        :      500
  Maximum-Amplitude        :      7000
  Minimum-Slope            :      20
  Maximum-Slope            :      1000
  Slope-Direction          :      Negative
  Selection-Policy         :      Max-Amplitude
////////////////////////////////////////////////////////////////////////////////////////////
```

7. Muscle Waveform Model Specification

The muscle waveform model contains two patterns, up-start-muscle and down-start-muscle. Two characteristic line segments, pm and nm, are concatenated.

```
//////////////////////////////////////////////////////////////////////////////////////////
//          File Name: muscle.wave                                          //
//          Date: October 31, 1990.                                         //
//          Comment:  this is the waveform model                            //
//                    specification file for muscle activity.              //
//////////////////////////////////////////////////////////////////////////////////////////
  Waveform-Type            :     Bursts
  Waveform-Name            :     muscle
  Interval-Balance-Rate    :     1
  Amplitude-Change-Rate    :     1
  Number-Of-Wave-Models    :     2
//////////////////////////////////////////////////////////////////////////////////////////
//          Waveform Model for Uprising Spikes                              //
//////////////////////////////////////////////////////////////////////////////////////////
//          pm --------- the positive line segment.                         //
//          nm --------- the negative line segment.                         //
//////////////////////////////////////////////////////////////////////////////////////////
  PatternDefinition
   Waveform-Model-Name                :   up-start-muscle
   Minimum-Total-Duration             :   10   // (unit:msec)
   Maximum-Total-Duration             :   180  // (unit:msec)
   Average-Duty-Ratio                 :   1    // (unit:%)
   Sequential-Segment-Pattern         :   <pm><nm><pm><pm><nm>
                                          <pm><nm><pm><nm>
   Minimum-Single-Wave-Period         :   <10><10><10><10><10><10>
                                          <10><10>     // (unit:msec)
   Maximum-Single-Wave-Period         :   <50><50><50><50><50><50>
                                          <50><50>     // (unit:msec)
   Minimum-Single-Wave-Duty-Ratio     :   <1><1><1><1><1><1>
                                          <1><1>       // (unit:%)
  EndDefinition
  PatternDefinition
   Waveform-Model-Name                :   up-start-muscle
   Minimum-Total-Duration             :   10   // (unit:msec)
   Maximum-Total-Duration             :   180  // (unit:msec)
   Average-Duty-Ratio                 :   1    // (unit:%)
   Sequential-Segment-Pattern         :   <pm><nm><pm><pm><nm>
                                          <pm><nm><pm><nm>
   Minimum-Single-Wave-Period         :   <10><10><10><10><10><10>
                                          <10><10>     // (unit:msec)
   Maximum-Single-Wave-Period         :   <50><50><50><50><50><50>
                                          <50><50>     // (unit:msec)
   Minimum-Single-Wave-Duty-Ratio     :   <1><1><1><1><1><1>
                                          <1><1>       // (unit:%)
  EndDefinition
//////////////////////////////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////////////////////////////////
//              File Name: pm.seg                                                        //
//              Date: October 31, 1990.                                                  //
//              Comment:   this is the line segment model                                //
//                         for muscle activity detection.                                //
////////////////////////////////////////////////////////////////////////////////////////////
//              Segment Name        :     pm                                             //
//              Time unit           :     millisec                                       //
//              Amplitude unit      :     sVolt (0.1 uVolt)                              //
//              Slope unit          :     0.1 sVolt / millisecond.                       //
//              In each Contraction-Rate, 50% is full range.                             //
//              Signal-To-Noise-Ratio is a signal / noise.                               //
////////////////////////////////////////////////////////////////////////////////////////////
  Upper-Contraction-Rate   :     10
  Lower-Contraction-Rate   :     10
  Minimum-Period           :     4
  Maximum-Period           :     8
  Minimum-Amplitude        :     100
  Maximum-Amplitude        :     10000
  Minimum-Slope            :     50
  Maximum-Slope            :     6000
  Slope-Direction          :     Positive
  Selection-Policy         :     Max-Amplitude
////////////////////////////////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////////////////////////////////
//              File Name: nm.seg                                                        //
//              Date: October 31, 1990.                                                  //
//              Comment:   this is the line segment model                                //
//                         for muscle activity detection.                                //
////////////////////////////////////////////////////////////////////////////////////////////
//              Segment Name        :     nm                                             //
//              Time unit           :     millisec                                       //
//              Amplitude unit      :     sVolt (0.1 uVolt)                              //
//              Slope unit          :     0.1 sVolt / millisecond.                       //
//              In each Contraction-Rate, 50% is full range.                             //
//              Signal-To-Noise-Ratio is a signal / noise.                               //
////////////////////////////////////////////////////////////////////////////////////////////
  Upper-Contraction-Rate   :     10
  Lower-Contraction-Rate   :     10
  Minimum-Period           :     4
  Maximum-Period           :     8
  Minimum-Amplitude        :     100
  Maximum-Amplitude        :     10000
  Minimum-Slope            :     50
  Maximum-Slope            :     6000
  Slope-Direction          :     Negative
  Selection-Policy         :     Max-Amplitude
////////////////////////////////////////////////////////////////////////////////////////////
```

APPENDIX B
TASK SPECIFICATION FOR CONTEXTUAL ANALYSIS

The knowledge-base contextual analysis system is configured according to the task specification files provided by the user when the system is initialized. An overall task specification file contains the description of the tasks to be performed by the coordinating rule-based system, while specialized task specification files contain the descriptions of the tasks to be performed by specialized rule-based systems. The overall task specification file has the extension "mcm," while the specialized task specification files have the extension "mcd."

In the specification files, "//" and a combination of "/*" and "*/" are used for comment. The time unit is always msec.

1. Overall Task Specification

This specification describes the hypothesis-confirmation process for the elimination of false positive detections during the epileptic spike detection. Spatial and temporal contextual clues obtained from eight EEG channels are analyzed to eliminate false positive detections generated by the spike waveform detector. The Coordinating Rule-Based System (CRBS) supervises a main Specialized Rule-Based System (SRBS), "espike," and a veto SRBS, "artifact," and two supporting SRBS, "slow" and "edischarge," and four conflicting SRBSes, "malpha," "msigma," "meyemove," and "artifact." The parameters "Min-Focus-Constitute-Events," "Min-Scrutiny-Synch-Events," and "Scrutiny-Policy" are reserved for epileptic focus estimation and reprocessing, which will be later expanded.

```
/////////////////////////////////////////////////////////////////////////////////////////
//              File Name: espike.mcm                                                  //
//              Date: November 15, 1990.                                               //
//              Comment: Overall task specification file.                              //
/////////////////////////////////////////////////////////////////////////////////////////

 // The event which we are primarily interested in.
 Main-Event                 :       espike

 // The events which can veto or contend against the main event.
 Veto-Events                :       <artifact>
 Conflicting-Events         :       <malpha><msigma><meyemove><artifact>

 // The events which can support the main event.
 Supporting-Events          :       <edischarge><slow>
 Supporting-Intervals       :       <250><1200>

 // The latency and Queue Lingering Period for detection process.
 Detect-Latency-Period      :       10000
 Queue-Lingering-Period     :       10000

 // The interchannel overlap duration for being spatially related.
 Min-Interchannel-Overlap-Duration    :   30

 // Reserved parameters for estimating epileptic focus and reprocessing.
 // The policy for Scrutiny are either Loose, Moderate, or Strict.
 Min-Focus-Constitute-Events          :   2
 Scrutiny-Policy                      :   Moderate
 Min-Scrutiny-Synch-Events            :   2

/////////////////////////////////////////////////////////////////////////////////////////
//              Inference Engine Parameters                                            //
/////////////////////////////////////////////////////////////////////////////////////////

 Inference-Goal             :       (is this espike)
 Counter-Goal               :       (is-not this espike)

/////////////////////////////////////////////////////////////////////////////////////////
//              Rule Declaration                                                       //
/////////////////////////////////////////////////////////////////////////////////////////
Domain-Rule-Specification

    RULE-NAME Rule00
    IF              ((spatial-support ?x strong)
                    (directly-supporting ?y ?x)
                    (existence ?y genuine))
    THEN            (is ?x espike)
    END-RULE

    RULE-NAME Rule01
    IF              ((spatial-support ?x strong)
                    (indirectly-supporting ?y ?x)
                    (existence ?y genuine))
```

```
THEN          (is ?x espike)
END-RULE

RULE-NAME Rule02
IF            ((spatial-support ?x strong)
               (directly-supporting ?y ?x)
               (existence ?y unclear))
THEN          (is ?x espike)
END-RULE

RULE-NAME Rule10
IF            ((spatial-support ?x normal)
               (directly-supporting ?y ?x)
               (existence ?y genuine))
THEN          (is ?x espike)
END-RULE

RULE-NAME Rule11
IF            ((spatial-support ?x normal)
               (indirectly-supporting ?y ?x)
               (existence ?y genuine))
THEN          (is ?x espike)
END-RULE

RULE-NAME Rule12
IF            ((spatial-support ?x normal)
               (directly-supporting ?y ?x)
               (existence ?y unclear))
THEN          (is ?x espike)
END-RULE

RULE-NAME Rule20
IF            ((spatial-support ?x weak)
               (directly-supporting ?y ?x)
               (existence ?y genuine))
THEN          (is ?x espike)
END-RULE

RULE-NAME Rule21
IF            ((spatial-support ?x weak)
               (indirectly-supporting ?y ?x)
               (existence ?y genuine))
THEN          (is ?x espike)
END-RULE

RULE-NAME Rule22
IF            ((spatial-support ?x weak)
               (directly-supporting ?y ?x)
               (existence ?y unclear))
THEN          (is ?x espike)
END-RULE

RULE-NAME Rule30
IF            ((spatial-support ?x bad)
```

```
                  (directly-supporting edischarge ?x)
                  (existence edischarge genuine)
                  (directly-supporting slow ?x)
                  (existence slow genuine))
THEN              (is ?x espike)
END-RULE

RULE-NAME Rule31
IF                ((spatial-support ?x bad)
                  (indirectly-supporting edischarge ?x)
                  (existence edischarge genuine)
                  (directly-supporting slow ?x)
                  (existence slow genuine))
THEN              (is ?x espike)
END-RULE

RULE-NAME Rule32
IF                ((spatial-support ?x bad)
                  (directly-supporting edischarge ?x)
                  (existence edischarge genuine)
                  (indirectly-supporting slow ?x)
                  (existence slow genuine))
THEN              (is ?x espike)
END-RULE

RULE-NAME Rule33
IF                ((spatial-support ?x bad)
                  (indirectly-supporting edischarge ?x)
                  (existence edischarge genuine)
                  (indirectly-supporting slow ?x)
                  (existence slow genuine))
THEN              (is ?x espike)
END-RULE

RULE-NAME RULE200
IF                ((directly-contending ?x malpha)
                  (existence malpha genuine))
THEN              (is-not ?x espike)
END-RULE

RULE-NAME RULE201
IF                ((directly-contending ?x msigma)
                  (existence msigma genuine))
THEN              (is-not ?x espike)
END-RULE

RULE-NAME RULE202
IF                ((directly-contending ?x artifact)
                  (existence artifact genuine))
THEN              (is-not ?x espike)
END-RULE

RULE-NAME RULE210
IF                ((indirectly-contending ?x malpha)
```

```
                    (existence malpha genuine))
 THEN               (is-not ?x espike)
 END-RULE

RULE-NAME RULE220
 IF                 ((spatial-support ?x bad)
                    (has-no ?x temporal-support))
 THEN               (is-not ?x espike)
 END-RULE

RULE-NAME RULE230
 IF                 ((occur-in-eyechannels ?x all)
                    (directly-contending ?x meyemove)
                    (existence meyemove genuine))
 THEN               (is-not ?x espike)
 END-RULE

End-Rule-Specification
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

2. Specialized Task Specification for detecting Epileptic Spikes

This specification file specifies the task of detecting epileptic spikes from multiple channels of EEG. The espike SRBS serves as a main SRBS, which can alone determine the final official view of the hypothesis-confirmation process. Here MCE stands for Multi-Channel Event.

```
//////////////////////////////////////////////////////////////////////////////////////////
//              File Name : espike.mcd                                                  //
//              Date: November 15, 1990.                                                //
//              Comment: Specialized task specification file for espike.              //
//////////////////////////////////////////////////////////////////////////////////////////

 // The multichannel event to detect.
 Multi-Channel-Event-Name:        espike

 // Waveform event that constitute the MCE.
 Constituent-Event-Type    :      spike

 // The number of MCE models.
 Number-Of-Event-Models  :        1

 // MCE model description always starts with the following line.
 MCE-Model-Specification
    Event-Model-Name                    :  referential-espike
    Supporting-Precursors               :  None
    Precursor-Zone-Length               :  None
    Supporting-Postcursors              :  <discharge><delta>
    Postcursor-Zone-Length              :  <250><1000>
    Conflicting-Events                  :  <muscle><alpha><sigma>
    Conflicting-Overlap-Duration        :  <30><30><30>
    Min-Interchannel-Overlap-Duration   :  30
 End-Model-Specification

//////////////////////////////////////////////////////////////////////////////////////////
//              Inference Engine Parameters                                             //
//////////////////////////////////////////////////////////////////////////////////////////

 Inference-Goal          :       (is this espike)
 Counter-Goal            :       (is-not this espike)

//////////////////////////////////////////////////////////////////////////////////////////
//              Rule Declaration                                                        //
//////////////////////////////////////////////////////////////////////////////////////////
Domain-Rule-Specification

    RULE-NAME    Rule00
    IF           ((spatial-support ?x strong)
                 (has-no ?x temporal-conflict)
```

```
                (has-supporting-postcursor ?x discharge)
                (has-supporting-postcursor ?x delta))
   THEN         (is ?x espike)
   END-RULE

   RULE-NAME    Rule10
   IF           ((spatial-support ?x normal)
                (has-no ?x temporal-conflict)
                (has-supporting-postcursor ?x discharge)
                (has-supporting-postcursor ?x delta)
   THEN         (is ?x espike)
   END-RULE

   RULE-NAME    Rule20
   IF           ((spatial-support ?x weak)
                (has-no ?x temporal-conflict)
                (has-supporting-postcursor ?x discharge)
                (has-supporting-postcursor ?x delta)
                (has ?x temporal-support))
   THEN         (is ?x espike)
   END-RULE

   RULE-NAME    Rule100
   IF           ((spatial-support ?x bad)
                (has-conflicting-contender ?x ?y))
   THEN         (is-not ?x espike)
   END-RULE

   RULE-NAME    Rule101
   IF           ((has-conflicting-contender ?x muscle))
   THEN         (is-not ?x espike)
   END-RULE
End-Rule-Specification
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

3. Specialized Task Specification for detecting Discharge Waves

This specification file specifies the task of detecting discharge waves from multichannel EEG, which follow right after the spike. The edischarge SRBS serves as a supporting SRBS, which provides contextual evidences supporting the given hypothesis.

```
////////////////////////////////////////////////////////////////////////////////////////
//              File Name : edischarge.mcd                                           //
//              Date : November 15, 1990.                                            //
//              Comment : Specialized task specification file for edischarge.        //
////////////////////////////////////////////////////////////////////////////////////////

 // The multichannel event to detect.
 Multi-Channel-Event-Name:        edischarge
 // Waveform event that constitute the MCE.
 Constituent-Event-Type     :        discharge

 // The number of MCE models.
 Number-Of-Event-Models  :        1

 // MCE model description always starts with the following line.
 MCE-Model-Specification
    Event-Model-Name                    :   multiple-channel-discharge
    Supporting-Precursors               :   <spike>
    Precursor-Zone-Length               :   <60>
    Supporting-Postcursors              :   <delta>
    Postcursor-Zone-Length              :   <600>
    Conflicting-Events                  :   <alpha><eyemove>
    Conflicting-Overlap-Duration        :   <75><80>
    Min-Interchannel-Overlap-Duration :   80
 End-Model-Specification

////////////////////////////////////////////////////////////////////////////////////////
//              Inference Engine Parameters                                          //
////////////////////////////////////////////////////////////////////////////////////////

 Inference-Goal             :        (is this edischarge)
 Counter-Goal               :        (is-not this edischarge)

////////////////////////////////////////////////////////////////////////////////////////
//              Rule Declaration                                                     //
////////////////////////////////////////////////////////////////////////////////////////
Domain-Rule-Specification
    RULE-NAME     Rulex0
    IF            ((has-no ?x temporal-conflict))
    THEN          (is ?x edischarge)
    END-RULE

    RULE-NAME     Rule00
    IF            ((spatial-support ?x strong))
```

```
THEN          (is ?x edischarge)
END-RULE

RULE-NAME     Rule10
IF            ((spatial-support ?x normal))
THEN          (is ?x edischarge)
END-RULE

RULE-NAME     Rule20
IF            ((spatial-support ?x weak)
              (has-supporting-postcursor ?x delta))
THEN          (is ?x edischarge)
END-RULE

RULE-NAME     Rule30
IF            ((spatial-support ?x bad)
              (has-supporting-postcursor ?x spike)
              (has-supporting-postcursor ?x delta))
THEN          (is ?x edischarge)
END-RULE


RULE-NAME     Rule100
IF            ((has-conflicting-contender ?x alpha))
THEN          (is-not ?x edischarge)
END-RULE

End-Rule-Specification
////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

4. Specialized Task Specification for detecting Alpha Spindles

This specification file specifies the task of detecting alpha spindles from multichannel EEG. The malpha SRBS serves as a conflicting SRBS, which provides contextual evidences conflicting with the given hypothesis.

```
///////////////////////////////////////////////////////////////////////////////////////////
//              File Name: malpha.mcd                                                   //
//              Date: November 15, 1990.                                                //
//              Comment: Specialized task specification file for malpha.               //
///////////////////////////////////////////////////////////////////////////////////////////

  // The multichannel event to detect.
  Multi-Channel-Event-Name :        malpha

  // Waveform event that constitute the MCE.
  Constituent-Event-Type    :        alpha

  // The number of MCE models.
  Number-Of-Event-Models  :        1

  // MCE model description always starts with the following line.
  MCE-Model-Specification
     Event-Model-Name                    :   multiple-channel-alpha
     Supporting-Precursors               :   <alpha>
     Precursor-Zone-Length               :   <5000>
     Supporting-Postcursors              :   <alpha>
     Postcursor-Zone-Length              :   <5000>
     Conflicting-Events                  :   <sigma><muscle>
     Conflicting-Overlap-Duration        :   <150><150>
     Min-Interchannel-Overlap-Duration :   150
  End-Model-Specification

///////////////////////////////////////////////////////////////////////////////////////////
//              Inference Engine Parameters                                             //
///////////////////////////////////////////////////////////////////////////////////////////

  Inference-Goal            :        (is this malpha)
  Counter-Goal              :        (is-not this malpha)

///////////////////////////////////////////////////////////////////////////////////////////
//              Rule Declaration                                                        //
///////////////////////////////////////////////////////////////////////////////////////////
Domain-Rule-Specification
    RULE-NAME    Rule00
    IF           ((spatial-support ?x strong))
    THEN         (is ?x malpha)
    END-RULE

    RULE-NAME    Rule10
```

```
IF            ((spatial-support ?x normal))
THEN          (is ?x malpha)
END-RULE

RULE-NAME     Rule20
IF            ((spatial-support ?x weak))
THEN          (is ?x malpha)
END-RULE

RULE-NAME     Rule30
IF            ((spatial-support ?x bad))
THEN          (is ?x malpha)
END-RULE


RULE-NAME     Rule100
IF            ((spatial-support ?x bad)
              (has-conflicting-contender ?x ?y)
              (has-no ?x temporal-support))
THEN          (is-not ?x malpha)
END-RULE

End-Rule-Specification
/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

5. Specialized Task Specification for detecting Sigma Spindles

This specification file specifies the task of detecting sigma spindles from multichannel EEG. In this study, the msigma SRBS serves as a conflicting SRBS, which provides contextual evidences conflicting with the given hypothesis.

```
/////////////////////////////////////////////////////////////////////////////////////
//              File Name: msigma.mcd                                              //
//              Date: November 15, 1990.                                           //
//              Comment: Specialized task specification file for msigma.           //
/////////////////////////////////////////////////////////////////////////////////////

 // The multichannel event to detect.
 Multi-Channel-Event-Name:          msigma

 // Waveform event that constitute the MCE.
 Constituent-Event-Type    :        sigma

 // The number of MCE models.
 Number-Of-Event-Models  :          1

 // MCE model description always starts with the following line.
 MCE-Model-Specification
    Event-Model-Name                 :   multiple-channel-sigma
    Supporting-Precursors            :   <sigma>
    Precursor-Zone-Length            :   <10000>
    Supporting-Postcursors           :   <sigma>
    Postcursor-Zone-Length           :   <10000>
    Conflicting-Events               :   <alpha><muscle>
    Conflicting-Overlap-Duration     :   <150><150>
    Min-Interchannel-Overlap-Duration :  100
 End-Model-Specification

/////////////////////////////////////////////////////////////////////////////////////
//              Inference Engine Parameters                                        //
/////////////////////////////////////////////////////////////////////////////////////

 Inference-Goal          :     (is this msigma)
 Counter-Goal            :     (is-not this msigma)

/////////////////////////////////////////////////////////////////////////////////////
//              Rule Declaration                                                   //
/////////////////////////////////////////////////////////////////////////////////////
Domain-Rule-Specification

    RULE-NAME    Rule00
    IF           ((spatial-support ?x strong))
    THEN         (is ?x msigma)
    END-RULE
```

```
RULE-NAME    Rule10
IF           ((spatial-support ?x normal))
THEN         (is ?x msigma)
END-RULE

RULE-NAME    Rule20
IF           ((spatial-support ?x weak))
THEN         (is ?x msigma)
END-RULE

RULE-NAME    Rule30
IF           ((spatial-support ?x bad))
THEN         (is ?x msigma)
END-RULE

RULE-NAME    Rule100
IF           ((spatial-support ?x bad)
             (has-conflicting-contender ?x ?y)
             (has-no ?x temporal-support))
THEN         (is-not ?x msigma)
END-RULE

End-Rule-Specification
////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

6. Specialized Task Specification for detecting Muscle Artifacts

This specification file specifies the task of detecting muscle artifacts from multichannel EEG. In this study, the artifact SRBS serves as a conflicting SRBS, which provides contextual evidences conflicting with the given hypothesis.

```
////////////////////////////////////////////////////////////////////////////
//            File Name: artifact.mcd                                     //
//            Date: November 15, 1990.                                    //
//            Comment: Specialized task specification file for artifact. //
////////////////////////////////////////////////////////////////////////////

 // The multichannel event to detect.
 Multi-Channel-Event-Name:        artifact

 // Waveform event that constitute the MCE.
 Constituent-Event-Type     :     muscle

 // The number of MCE models.
 Number-Of-Event-Models  :        1

 // MCE model description always starts with the following line.
 MCE-Model-Specification
    Event-Model-Name                   :  multiple-channel-muscle
    Supporting-Precursors              :  <muscle>
    Precursor-Zone-Length              :  <1000>
    Supporting-Postcursors             :  <muscle>
    Postcursor-Zone-Length             :  <1000>
    Conflicting-Events                 :  None
    Conflicting-Overlap-Duration       :  None
    Min-Interchannel-Overlap-Duration  :  300
 End-Model-Specification

////////////////////////////////////////////////////////////////////////////
//            Inference Engine Parameters                                 //
////////////////////////////////////////////////////////////////////////////

 Inference-Goal         :     (is this artifact)
 Counter-Goal           :     (is-not this artifact)

////////////////////////////////////////////////////////////////////////////
//            Rule Declaration                                            //
////////////////////////////////////////////////////////////////////////////
Domain-Rule-Specification

    RULE-NAME    Rule00
    IF           ((spatial-support ?x strong))
    THEN         (is ?x artifact)
    END-RULE
```

```
RULE-NAME    Rule10
IF           ((spatial-support ?x normal))
THEN         (is ?x artifact)
END-RULE

RULE-NAME    Rule20
IF           ((spatial-support ?x weak))
THEN         (is ?x artifact)
END-RULE

RULE-NAME    Rule30
IF           ((spatial-support ?x bad)
             (has-supporting-precursor ?x muscle))
THEN         (is ?x artifact)
END-RULE

RULE-NAME    Rule31
IF           ((spatial-support ?x bad)
             (has-supporting-postcursor ?x muscle))
THEN         (is ?x artifact)
END-RULE

End-Rule-Specification
/////////////////////////////////////////////////////////////////////////////////////////////////////////
```

7. Specialized Task Specification for detecting Eye-Movements

This specification file specifies the task of detecting eye-movements from multichannel EEG. In this study, the meyemove SRBS serves as a conflicting SRBS, which provides contextual evidences conflicting with the given hypothesis.

```
//////////////////////////////////////////////////////////////////////////////////////////////
//          File Name: meyemove.mcd                                         //
//          Date: November 15, 1990.                                        //
//          Comment: Specialized task specification file for meyemove.    //
//////////////////////////////////////////////////////////////////////////////////////////////

 // The multichannel event to detect.
 Multi-Channel-Event-Name :      meyemove

 // Waveform event that constitute the MCE.
 Constituent-Event-Type    :      eyemove

 // The number of MCE models.
 Number-Of-Event-Models  :       1

 // MCE model description always starts with the following line.
 MCE-Model-Specification
    Event-Model-Name                    :  multiple-channel-eyemove
    Supporting-Precursors               :  <eyemove>
    Precursor-Zone-Length               :  <2000>
    Supporting-Postcursors              :  <eyemove>
    Postcursor-Zone-Length              :  <2000>
    Conflicting-Events                  :  None
    Conflicting-Overlap-Duration        :  None
    Min-Interchannel-Overlap-Duration   :  300
 End-Model-Specification

//////////////////////////////////////////////////////////////////////////////////////////////
//          Inference Engine Parameters                                     //
//////////////////////////////////////////////////////////////////////////////////////////////

 Inference-Goal          :       (is this meyemove)
 Counter-Goal            :       (is-not this meyemove)

//////////////////////////////////////////////////////////////////////////////////////////////
//          Rule Declaration                                                //
//////////////////////////////////////////////////////////////////////////////////////////////
Domain-Rule-Specification

    RULE-NAME    Rule00
    IF           ((spatial-support ?x strong)
                 (occur-in-eyechannels ?x all))
    THEN         (is ?x meyemove)
    END-RULE
```

```
RULE-NAME   Rule10
IF          ((spatial-support ?x normal)
            (occur-in-eyechannels ?x all))
THEN        (is ?x meyemove)
END-RULE

RULE-NAME   Rule20
IF          ((spatial-support ?x weak)
            (occur-in-eyechannels ?x all))
THEN        (is ?x meyemove)
END-RULE

RULE-NAME   Rule30
IF          ((spatial-support ?x bad)
            (occur-in-eyechannels ?x all)
            (has-supporting-precursor ?x ?y))
THEN        (is ?x meyemove)
END-RULE

RULE-NAME   Rule31
IF          ((spatial-support ?x bad)
            (occur-in-eyechannels ?x all)
            (has-supporting-postcursor ?x ?y))
THEN        (is ?x meyemove)
END-RULE

RULE-NAME Rule100
IF          ((occur-in-eyechannels ?x some))
THEN        (is-not ?x meyemove)
END-RULE

RULE-NAME   Rule101
IF          ((occur-in-eyechannels ?x none))
THEN        (is-not ?x meyemove)
END-RULE

End-Rule-Specification
///////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

8. Specialized Task Specification for detecting Slow Waves

This specification file specifies the task of detecting slow waves from multichannel EEG. In this study, the slow SRBS serves as a supporting SRBS, which provides contextual evidences that support the given hypothesis.

```
/////////////////////////////////////////////////////////////////////////////////
//          File Name: slow.mcd                                                //
//          Date: November 15, 1990.                                           //
//          Comment: Specialized task specification file for slow wave.       //
/////////////////////////////////////////////////////////////////////////////////

 // The multichannel event to detect.
 Multi-Channel-Event-Name:       slow

 // Waveform event that constitute the MCE.
 Constituent-Event-Type    :     delta

 // The number of MCE models.
 Number-Of-Event-Models  :       1

 // MCE model description always starts with the following line.
 MCE-Model-Specification
    Event-Model-Name                  :  multiple-channel-slow
    Supporting-Precursors             :  None
    Precursor-Zone-Length             :  None
    Supporting-Postcursors            :  None
    Postcursor-Zone-Length            :  None
    Conflicting-Events                :  <eyemove>
    Conflicting-Overlap-Duration      :  <200>
    Min-Interchannel-Overlap-Duration :  200
 End-Model-Specification

/////////////////////////////////////////////////////////////////////////////////
//          Inference Engine Parameters                                        //
/////////////////////////////////////////////////////////////////////////////////

 Inference-Goal          :      (is this slow)
 Counter-Goal            :      (is-not this slow)

/////////////////////////////////////////////////////////////////////////////////
//          Rule Declaration                                                   //
/////////////////////////////////////////////////////////////////////////////////
Domain-Rule-Specification

    RULE-NAME    Rulex0
    IF           ((has-no ?x temporal-conflict))
    THEN         (is ?x slow)
    END-RULE
    RULE-NAME    Rule00
```

```
IF            ((spatial-support ?x strong))
THEN          (is ?x slow)
END-RULE

RULE-NAME     Rule10
IF            ((spatial-support ?x normal))
THEN          (is ?x slow)
END-RULE

RULE-NAME     Rule20
IF            ((spatial-support ?x weak))
THEN          (is ?x slow)
END-RULE

RULE-NAME     Rule30
IF            ((spatial-support ?x bad))
THEN          (is ?x slow)
END-RULE

RULE-NAME     Rule100
IF            ((occur-in-eyechannels ?x all)
              (has-conflicting-contender ?x ?y))
THEN          (is-not ?x slow)
END-RULE

End-Rule-Specification
////////////////////////////////////////////////////////////////////////////////////////////////////////
```

REFERENCES

[Al83a] Allen, J.F., "Maintaining Knowledge about Temporal Intervals," Communications of the ACM, vol. 26, 11, pp. 832-843, Nov. 1983.

[Al83b] Allen, J.F. and Hayes, P.J., "A Common Sense Theory of Time," in Proceedings of IJCAI, pp. 528-531, 1985.

[Ba89] Barr, A., Cohen, P.R., and Feigenbaum, E.A., The Handbook of Artificial Intelligence, Addison-Wesley Publishing Company, Reading Massachusetts, 1989.

[Bo86] Booch, G., "Object-Oriented Development," IEEE Trans. Software Eng., vol. SE-12, no. 2, pp. 211-221, Feb. 1986.

[Ch74] Chatrian, G.E., Bergamini, L., Dondey, M., Klass, D.W., Lennox-Buchthal, M., and Petersen, I., "A Glossary of Terms most commonly used by Clinical Electroencephalographers," Electroencephalography and Clinical Neurophysiology, vol 37, pp. 538-548, Elsevier, Amsterdam, 1974.

[Co87] Cox, B.J., Object-Oriented Programming - An Evolutionary Approach, Addison-Wesley, Reading, Massachusetts, 1987.

[Du89] Duffy, F.H., Iyer, V.G., Surwillo, W.W., Clinical Electroencephalography and Topographic Brain Mapping, Springer-Verlag, New York, 1989.

[Fr80] Frost, J.D., Hilman, C.E. and Kellaway, P., "Automatic Interpretation of EEG: Analysis of Background Activity," Computers and Biomedical Research, vol. 13, pp. 242-257, Academic Press, London, 1980.

[Ge75] Gevins, A.S., Yeager, C.L., Diamond, S.L., Spire, J.P., Zeitlin, G.M. and Gevins, A.H., "Automated Analysis of the Electrical Activity of the Human-Brain (EEG): a progress report," Proc. IEEE, 63, pp. 1382-1399, 1975.

[Ge76] Gevins, A.S., Yeager, C.L., Diamond, S.L., Zeitlin, G.M., Spire, J.P. and Gevins, A.H., "Sharp Transient Analysis and Threshold Linear Coherence Spectra of Paroxysmal EEGs," in Quantitative Analytic Studies in Epilepsy, Kellaway, P. and Petersen, I., Eds., Raven Press, New York, pp. 463, 1976.

[Gl86] Glover, J.R., Ktonas, P.Y., Raghavan, N., Urunuela, J.M., Velamuri, S.S., and Reilly, E.L., "A Multichannel Signal Processor for the Detection of Epileptogenic Sharp Transients in the EEG," IEEE Trans. Biomedical Eng., vol. BME-33, No. 12, pp. 1121-1128, Dec. 1986.

[Gl89] Glover, J.R., Raghavan, Ktonas, P.Y., and Frost, J.D., "Context-Based Automated Detection of Epileptogenic Sharp Transients in the EEG: Elimination of False Positives," IEEE Trans. Biomedical Eng., vol. 36, No. 5, pp. 519-527, May 1989.

[Go73] Goldberg, P., Samson-Dollfus, D. and Gremy, F., "An Approach to Automatic Pattern Recognition of the Electroencephalogram: Background Rhythm and Paroxysmal Elements," Meth. Inform. Med., 12, pp. 155-163, 1973.

[Got86] Gotman, J., "Computer Analysis of the EEG in Epilepsy," Handbook of Electroencephalography and Clinical Neurophysiology, Revised Series, vol 2, Elsevier, Amsterdam, 1986.

[Gotm79a] Gotman, J. and Gloor, P., "Automatic Recognition and Quantification of Interictal Epileptic Activity in the Human Scalp EEG," Electroencephalography and Clinical Neurophysiology, vol. 41, pp. 513-529, Elsevier, Amsterdam, 1976.

[Gotm79b] Gotman, J., Ives, J.R., Gloor, P., "Automatic Recognition of Interictal Epileptic Activity in Prolonged EEG Recordings," Electroencephalography and Clinical Neurophysiology, vol. 46, pp. 510-520, Elsevier, Amsterdam, 1979.

[Ja85] Jansen, B.H., "Analysis of Biomedical Signals by means of Linear Modeling," CRC Critical Reviews in Biomedical Eng., vol. 12, Issue 4, pp. 343-392, 1985.

[Ja89] Jansen, B.H., Dawant, B.M., "Knowledge-Based Approach to Sleep EEG Analysis — A Feasibility Study," IEEE Trans. Biomedical Eng., vol. 36, No. 5, pp. 510-518, May 1989.

[Ko88] Koffler, D.J. and Gotman, J., "Automatic Detection of Spike-and-Wave Bursts in Ambulatory EEG Recordings," Electroencephalography and Clinical Neurophysiology, vol. 61, pp. 165-180, Elsevier, Amsterdam, 1985.

[Kt83] Ktonas, "Automatic Detection of Spike-and-Wave Bursts in Ambulatory EEG Recordings," Electroencephalography and Clinical Neurophysiology, vol. 61, pp. 165-180, Elsevier, Amsterdam, 1985.

[Ku88] Kuwahara, H., Higashi, H., Mizuki, Y., Matsunari, S., Tanaka, M. and Inanaga, K., "Automatic Real-time Analysis of Human Sleep Stages by an Interval Histogram method," Electroencephalography and Clinical Neurophysiology, vol 70, pp. 220-229, Elsevier, Amsterdam, 1988.

[Li80] Lim, A.J., and Winters, W.D., "A Practical Method for Automatic Real-time EEG Sleep State Analysis," IEEE Trans. Biomedical Eng., vol. BME-27, pp. 212-220, 1980.

[Lo75] Lopes da Silva, F.H., Dijk, A. and Smiths, H., "Detection of Nonstationarities in EEGs using the Autoregressive Model – an Application to EEGs of Epileptics," in CEAN – Computerized EEG Analysis, Kunkel, H. and Dolce, G. Eds., pp. 180-199, Gustav Fischer Verlag, Stuttgart, 1975.

[Lo77] Lopes da Silva, F.H., Van Hulten, K., Lommen, J.G., Storm van Leeuwen, W., Van Veelen, C.W.M., and Vliegenthart, W., "Automatic Detection and Localization of Epileptic Foci," Electroencephalography and Clinical Neurophysiology, vol 43, pp. 1-13, Elsevier, Amsterdam, 1977.

[Lo87] Lopes da Silva, F.H. and Storm van Leeuwen, W., "General Introduction," in Clinical Applications of Computer Analysis of EEG and other Neurophysiological Signals, EEG Handbook (Revised Series, vol. 2), F.H. Lopes da Silva, W. Storm van Leeuwen, and A. Remond, Eds. pp. 1-10, Elsevier, Amsterdam, 1987.

[Ma76] Ma, K.M., Celesia, G.G. and Birkemeier., "Cluster Analysis and Spike Detection in EEG," in Epileptogy, Janz, D., Eds., pp. 386-396, Georg Thieme, Stuttgart, 1976.

[Me88] Meyer, B., Object-Oriented Software Construction, Prentice Hall International (UK) Ltd., Hertfordshire, 1988.

[Ol83] Oliveira, P.G., Queiroz, C. and Silva, F.L., "Spike Detection based on a Pattern Recognition Approach using a Microcomputer," Electroencephalography and Clinical Neurophysiology, vol 56, pp. 97-103, Elsevier, Amsterdam, 1983.

[Op75] Oppenheim, A.V. and Shafer, R.W., Digital Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, 1975.

[Pa90] Park, S-H, Principe, J.C., Smith, J.R., Reid, S.A., "TDAT – Time Domain Analysis Tool for EEG Analysis," IEEE Trans. on Biomedical Engineering, vol. 37, No. 8, pp. 803-811, Aug. 1990.

[Par72] Parnas, D. L., "On the Criteria to be used in Decomposing Systems into Modules," Commun. ACM, Dec. 1972.

[Pav73] Pavlidis, T., "Waveform Segmentation Through Functional Approximation," IEEE Trans. Computers, vol. C-22, No. 7, pp. 689-697, Jul. 1973.

[Pr85] Principe, J. C. and Smith, J.R., "Automatic Detection of Spike-and-Wave Bursts," in Long-Term Monitoring in Epilepsy (EEG supl. No. 37), J. Gotman, J.R. Ives, and P. Gloor, Eds., pp. 115-131, Elsevier, Amsterdam, 1985.

[Re89] Reyes, A.C., Time Domain analysis Tool User's Manual, Sleep EEG Lab., University of Florida, Gainesville, Florida, 1989.

[Sh84] Shaw, M., "Abstraction Techniques in Modern Programming Languages," IEEE Trans. Software Eng., vol 1, no. 4, pp. 10, Oct. 1984.

[Sh90] Shaw, S.W. and Defigueiredo, R., "Structural Processing of Waveforms as Trees," IEEE Trans. Acoustics, Speech, and Signal Processing, vol. 38, no. 2, pp. 328-338, Feb. 1990.

[Sm74] Smith, J.R., "Automatic Analysis and Detection of EEG Spikes," IEEE Trans. on Biomedical Engineering, vol. BME-21, No. 1, pp. 1-7, Jan. 1974.

[Sm78] Smith, J.R., Karacan, I., and Yang, M., "Automated Analysis of the Human Sleep EEG," Waking Sleeping, vol. 2, pp. 75-82, 1978.

[Sm87] Smith, J. R., "Computer Analysis of Sleep Data," in Clinical Applications of Computer Analysis of EEG and other Neurophysiological Signals, EEG Handbook (revised series, vol. 2), F.H. Lopes da Silva, W. Storm van Leeuwen, and A. Remond, Eds. pp. 131-145, Elsevier, Amsterdam, 1987.

[St86] Stroustrup, B., The C++ Programming Language, Addison-Wesley, Reading, Massachusetts, 1987.

[Tr89] Trickey, H., "C++ versus Lisp: A Case Study," SIGPLAN Notices, vol. 23, no. 2, pp. 9-18, 1989.

[Tr90] Trahanias, P. and Skordalakis, E., "Syntactic Pattern Recognition of the ECG," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 7, pp. 648-657, Jul. 1990.

[Wi88] Wiener, R. S. and Pinson, L. J., An Introduction to Object-Oriented Programming and C++, Addison-Wesley, Reading, Massachusetts, 1988.

[Wh82] Whisler J.W., Remine, W.J., Leppik, I.E., Mclain, L.W. and Gumnit, R.J., "Machine Detection of Spike-Wave Activity in the EEG and its Accuracy compared with Visual Interpretation," Electroencephalography and Clinical Neurophysiology, vol 54, pp. 541-551, Elsevier, Amsterdam, 1982.

BIOGRAPHICAL SKETCH

Seung-Hun Park was born December 12, 1959, in JeonNahm, Korea. He earned a bachelor's degree in electrical engineering in February 1981 and a master's degree in control and instrumentation engineering in August 1984, from the Seoul National University, Seoul, Korea.

From January 1984 to August 1986, he was a member of the technical staff of Data Communication Section at the Electronics and Telecommunications Research Institute (ETRI), DaeJeon, Korea.

From August 1986 until the present, he has been a research assistant in the EEG Laboratory at the university of Florida.

I certify that I have read this study and that in my opinion it confirms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Jack R. Smith, Chair
Professor of Electrical Engineering

I certify that I have read this study and that in my opinion it confirms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Jose C. Principe, Cochair
Associate Professor
of Electrical Engineering

I certify that I have read this study and that in my opinion it confirms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Donald G. Childers

Donald G. Childers
Professor of Electrical Engineering

I certify that I have read this study and that in my opinion it confirms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Antonio A. Arroyo
Associate Professor
of Electrical Engineering

I certify that I have read this study and that in my opinion it confirms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Edward J. Hammond
Assistant Professor
of Neurology

I certify that I have read this study and that in my opinion it confirms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Steven A. Reid
Assistant Professor
of Neurological Surgery

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

December 1990

Herbert A. Bevis
for Winfred M. Phillips
Dean, College of Engineering

Madelyn M. Lockhart
Dean, Graduate School